第 16 卷第 6 期
1999 年 12 月

控制理论与应用
CONTROL THEORY AND APPLICATIONS

Vol. 16, No. 6
Dec., 1999

Genetic-Based Neurofuzzy Control for Complex Industrial Process *

Wang Yaonan and Zhang Changfan
(Department of Electricial Engineering, Hunan University · Changsha, 410082, P. R. China)

Abstract: This paper proposes an effective fuzzy neural network controller based on genetic algorithm (GA) and supervised gradient descent learning. The fuzzy network control processing can be viewed as a parallel neural network where each neuron represents a fuzzy membership function and each link represents the weight of a fuzzy rule, and it has two important characteristics of adaptation and learning. The effectiveness of the proposed scheme is illustrated through simulation and temperature control processes.

Key words: fuzzy net; learning algorithm; intelligent control

复杂工业过程的遗传模糊神经网络控制

王耀南 张昌凡
(湖南大学电气工程系·长沙,410082)

摘要: 本文提出一种基于遗传算法和监督学习方法的有效模糊神经网络控制.这种控制器采用并行处理的模糊推理网络,具有两个重要特点:自适应和学习性.所提方法经过仿真实验和温控验证表明控制性能良好.

关键词: 模糊神经网络; 学习算法; 智能控制

1 Introduction

Conventional fuzzy logic controllers generally require a certain reasonable set of fuzzy rules that integrate heuristics and intuition of human operators. The most important and difficult problem in the fuzzy logic controller design is how to obtain the proper control rules for a given plant. However, in the case of a system that has very complicated dynamic characteristics such as industrial rotary kiln, we will encounter significant difficulties to find the best fitted or at least reasonable fuzzy rules to control such a system.

Recently, fuzzy neural network control systems have been extensively studied[1~3]. The neural network ' s ability to produce arbitrarily nonlinear mappings has been demonstrated in various application studies. A well-trained network with such a capability could map input signals into adequate control actions for controlling complex dynamic systems. The neural network can learn control experience in some training courses by way of adequate updating for network parameters, and it has a greater tolerance in system uncertainty than traditional controllers do. The parallel distributed processing architecture enables the networks to achieve extremely fast computations. Fuzzy neural controllers, therefore, have great potentiality for controlling dynamic industrial kiln furnace processes.

In this paper, we propose a fuzzy logic network controller (FLNC) for industrial rotary kiln. The proposed controller can learn to control a complex system and adapt to a wide range of variations in plant parameters. To guarantee convergence and fast learning, the agjustment of the parameters in the proposed FLNC will be divided into two parts which are the IF (premise) part and THEN (consequence) part of the fuzzy logical rules. In the premise part, the shape of membership function can be optimized by means of a genetic algorithm. In the consequent part, the link weight of the fuzzy logical network is updated on-line using supervised gradient descent learning method.

2 Fuzzy neural network controller

2.1 Fuzzy logic control

In general, the dynamic behavior of a fuzzy logical controller is characterized by a set of linguistic control rules based on the knowledge of an expert. In this paper

* This research was supported by National 863 Foundation of China (863511845002).
Manuscript received Jul. 8, 1996, revised Feb. 23, 1998.

the simplified fuzzy reasoning in which the consequent parts are expressed by real numbers is employed in this method. When input variables are expressed by $x_j (j = 1, \cdots, m)$ and an output variable by y, the inference rules of the simplified fuzzy reasoning can be expressed by the following:

Rule i : If x_1 is A_{i1} and, $\cdots$, and x_m is A_{im} the y is w_i. (1)

Where $A_i (i = 1, \cdots, n)$ is the membership functions in the antecedent part, and w_i is a real number in the consequent part. The output of the simplified fuzzy reasoning, y can be derived by using the following equations:

$$\mu_i = \prod_{j=1}^{m} A_{ij}(x_j), \tag{2}$$

$$y = \frac{\sum_{i=1}^{n} \mu_i \cdot w_i}{\sum_{i=1}^{n} \mu_i}, \tag{3}$$

where μ_i is a membership value of i-th inference rule.

The proposed fuzzy neural feedback control system is shown in Fig. 1. The neural network-based, controller is a four-layer fuzzy logic network (FLNC) with two inputs (x_1 and x_2) and one output (control increment, u).

A schematic diagram of the proposed fuzzy logical network (FLNC) structure is shown in Fig. 2. Next, we shall indicate the signal propagation and the basic function of every node in each layer.

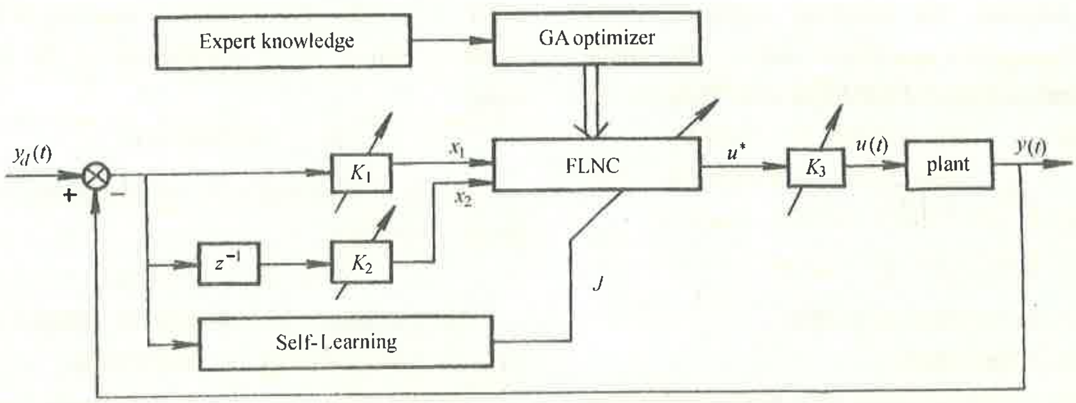

Fig. 1 The fuzzy logic net feedback control system

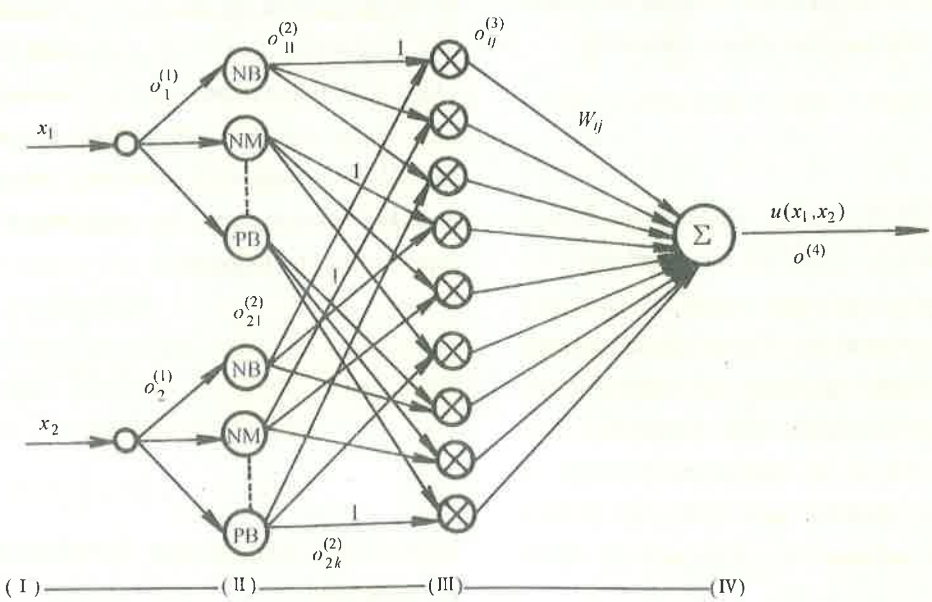

Fig. 2 Schematic diagram of a fuzzy logic network

Layer 1: input layer.

For the jth node of Layer 1, the network input and the network output are represented as:

$$I_j^{(1)} = W_{ij}^{(1)} \cdot x_i^{(1)}, \quad i = j, \quad o_j^{(1)} = f_j^{(1)}(I_j^{(1)}) = I_j^{(1)}, \tag{4}$$

where the weights $w_{ij}^{(1)}$ are assumed to be unity and $x_j^{(1)}$ represents the ith input to the jth node of Layer 1.

Layer 2: membership layer.

In this layer, each node performs a membership function. The Gaussian function is adopted here as a membership function.

Then

$$I_j^{(2)} = \mu_{Aij}(a_{ij}, b_{ij}) = -\frac{(x_i^{(2)} - a_{ij})^2}{(b_{ij})^2}, \qquad (5)$$
$$o_j^{(2)} = f_j^{(2)}(I_j^{(2)}) = \exp(I_j^{(2)}),$$

where a_{ij} and b_{ij} are, respectively, the center and the width of the Gaussian function in the jth term of the ith input linguistic variable $x_i^{(2)}$.

Layer 3: rule layer.

The links in this layer are used to implement the antecedent matching. The matching operation or the fuzzy AND aggregation operation is chosen as the simple product operation instead of MIN operation. Then, for the jth rule node

$$I_j^{(3)} = \prod_i^n W_{ij}^{(3)} \cdot x_i^{(3)}, \qquad (6)$$
$$o_j^{(3)} = f_j^{(3)}(I_j^{(3)}) = I_j^{(3)},$$

where $w_{ij}^{(3)}$ is also assumed to be unity.

Layer 4: output layer.

Since the overall net output is a linear combination of the consequences of all rules, the net input and output of the jth node in this layer are simply defined by

$$I_j^{(4)} = \sum_i^m W_{ij}^{(4)} \cdot x_i^{(4)}, \quad o_j^{(4)} = f_j^{(4)}(I_j^{(4)}) = I_j^{(4)}, \qquad (7)$$

where the link weight $W_{ij}^{(4)}$ is the output action strength of the output associated with the ith rule. Note that I_j, o_j and f_j are the summed net input, output and activation function of node j respectively. The above configuration shows that, by modifying the centers and widths of Layer 2 and the link weights of Layer 4, the membership function can be tuned and all the consequence strengths of fuzzy rules could be identified respectively. The learning process to train the proposed fuzzy logic network will be discussed in the following section.

3 Self-tuning method of fuzzy logic network

3.1 Learning method using genetic algorithm

The proposed method is to optimize the number of inference rules and the shapes (a_i, b_j) of the membership functions in the antecedent parts by a genetic algorithm.

A genetic algorithm (GA)[4] is a method to obtain an optimal solution by applying a theory of biological evolution. The most advantageous feature of the GA is a possibility of escaping from local optimum because of probabilistic operations such as crossover and mutation. In the GA, a solution candidate S_r which maximizes an objective function $F(S_r)$ called fitness, is searched. The solution candidate is expressed by the string, called individual, which is expressed by the following:

$$S_r = L_{r1} L_{r2} \cdots L_{rG},$$

where $L_{rg}(g = 1, \cdots, G)$ is a variable taking a value of either "1" or "0". For instance, an example of the individual S_r with $G = 13$ is expressed by the following string:

$$S_r = 1001000110011.$$

A set of individual, S, called population, is expressed as follows:

$$S = \{S_1, S_2, \cdots, S_R\}.$$

The procedures for obtaining the optimal solutions (a_i^*, b_i^*) using the GA are shown below:

1) The shapes (a_i, b_i) of membership function in the FLNC network are determined according to the string of the individual $S_r(t)$. The individuals $S_1, S_2, \cdots, S_N$, which constitute a population $S(t)$ of the 0-th generation $(t = 0)$ are determined by uniform random numbers.

2) The fitness $F(S_r)$ for each individual S_r is derived to determine a selection probability $P_{S_r}(t)$ which is expressed by the following:

$$P_{S_r}(t) = \frac{F(S_r(t))}{\sum_{r=1}^{N} F(S_r(t))}, \qquad (8)$$

$$F(S_r(t)) = 1/E, \qquad (9)$$

$$E = \frac{1}{2}\sum_{i=1}^{m}(\overline{U}_i - U_i^*), \qquad (10)$$

where $\overline{U}$ is the desired output, U_i^* is the actual output of the FLNC net.

3) Two individuals $S_i(t)$ and $S_j(t)$ are selected from the population $S(t)$ in accordance with the selection probabilities $P_{S_i}(t)$ and $P_{S_j}(t)$.

4) An operation called crossover is applied to the individuals $S_i(t)$ and $S_j(t)$. The crossover operation selects a boundary in the strings with probability of $1/(G-1)$, and exchanges the blocks of strings about the boundary. One of the two individuals produced by this operation is selected at random, and is nominated as the new individual $S'_k(t)$.

5) An operation called mutation is applied to $S'_k(t)$. By this, each element of the individual $S'_k(t)$ is reversed according to a mutation probability $P_m(0.05 < P_m < 0.1)$.

6) The number of newly produced individuals, k, is compared with the total number of individuals, N, and if $K < N$, k is increased by one, and steps 3) to 6) are repeated. Otherwise, the algorithm proceeds to the next step.

7) The new population, $S'(t) = \{S'_1(t), S'_2(t), \cdots, S'_R(t)\}$, produced in steps 3) to 6), is substituted into the population on the next generation $S(t+1)$.

8) The generation t is increased by one, and the steps 2) to 8) are repeated until a convergence of the population S is obtained.

An individual which has the highest fitness in the converged population is defined as the final solution (a_i^*, b_j^*).

3.2 Supervised gradient descent learning

A supervised learning law is used to tune on-line the link weights of Layer 4 in the FLNC. The basis of this algorithm is simply gradient descent. The derivation is the same as that of the back propagation learning law. By recursive applications of the chain rule, the error term for each layer is first calculated.

Let the cost function, J, for training pattern t be proportional to the sum of the square of the difference between the plant output $y(t)$ and the desired output $y_d(t)$, and let J be defined by

$$J = \frac{1}{2}[y_d(t) - y(t)]^2. \tag{11}$$

Then the gradient of error in Eq.(11) with respect to an arbitrary weighting vector $W^{(4)} \in \mathbb{R}^n$ becomes

$$\frac{\partial J}{\partial W^{(4)}} = e(t)\cdot\frac{\partial e(t)}{\partial W^{(4)}} = -e(t)\cdot\frac{\partial y(t)}{\partial W^{(4)}} = -e(t)\cdot\frac{\partial y(t)}{\partial u(t)}\cdot\frac{\partial u(t)}{\partial W^{(4)}} =$$

$$-e(t)\cdot y_u(t)\cdot\frac{\partial o^{(4)}}{\partial W^{(4)}} = -e(t)\cdot x^{(4)}\cdot y_u(t), \tag{12}$$

where

$$e(t) = y_d(t) - y(t)$$

is the error between the actual plant and the desired output,

$$y_u(t) = \partial y(t)/\partial u(t)$$

is the plant sensitivity. The plant sensitivity can be computed as follows:

$$\frac{\partial y(t)}{\partial u(t)} = \frac{y[u(t+1)] - y[u(t)]}{u(t+1) - u(t)}. \tag{13}$$

The weight can be adjusted by using a gradient method

$$W^{(4)}(t+1) = W^{(4)}(t) + \Delta W^{(4)}(t) = W^{(4)}(t) + \eta\left(-\frac{\partial J}{\partial W^{(4)}}\right), \tag{14}$$

where η is a learning rate.

4 System simulation and temperature control for rotary kiln

4.1 Examples of simulation

In system simulation, the FLNC is constructed by the 2-14-49-1 neuron. We used the 49 fuzzy control rules as shown in Table 1.

Table 1 Fuzzy control rules

$W_{ij}(0)$ x_1 \ x_2	NB	NM	NS	Z	PS	PM	PB
NB	-6.0	-6.0	-4.0	-6.0	-4.0	-4.0	-4.0
NM	-6.0	-4.0	-2.0	-4.0	-4.0	-4.0	-2.0
NS	-4.0	-4.0	-2.0	-2.0	-2.0	-0.0	-2.0
Z	-4.0	-2.0	0.0	0.0	2.0	4.0	-6.0
PS	-4.0	-2.0	4.0	2.0	2.0	2.0	4.0
PM	2.0	4.0	4.0	4.0	2.0	4.0	6.0
PB	4.0	4.0	4.0	6.0	4.0	6.0	6.0

The constants in this table are set as the initial connection weights $W_{ij}^{(4)}(0)$ for the FLNC. The central points of the fuzzy sets NB, NM, NS, Z, PS, PM, PB, $a_j(0)$ $(j = 1,2\cdots7)$ are $-6, -4, -2, 0, 2, 4, 6$, respectively. The width values of the membership function, $b_j(0)$ are all unity so as to equally allocate seven fuzzy sets on the range $[-6, 6]$, here, $b_j(0) = 2.5$.

Example 1

The controlled plant is a nonlinear system with a stochastic disturbance acting on it. The plant is in the form:

$$y(t) = \frac{y(t-1)e^{-y(t-1)} + u(t-1)}{1 + u(t-1)e^{-y(t-1)}} + \omega(t),$$

where $\omega(t)$ is a white noise with 0.15 standard variance.

The response of the plant is illustrated in Fig.3.

Example 2

The plant is a nonlinear system with time delay, i.e.

$$y(t) = \frac{y(t-1)y(t-2)y(t-3)y(t-4)u(t-4) - y(t-1)y(t-2)y(t-3)u(t-4) + u(t-5)}{1 + y^3(t-3) + y^3(t-4)},$$

where time delay $d = 4$, Fig.4 illustrates the step response curves of the FLNC control.

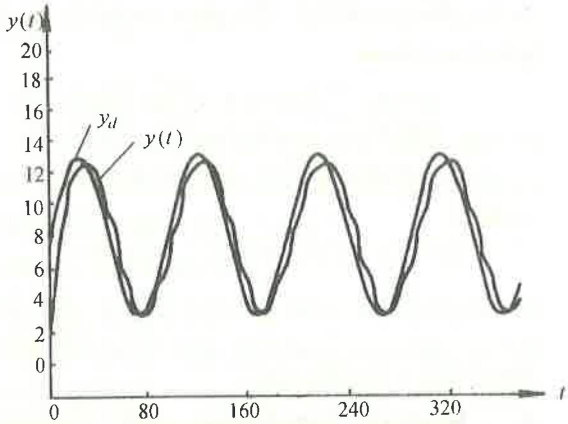

Fig. 3 Output of the plant for Example 1

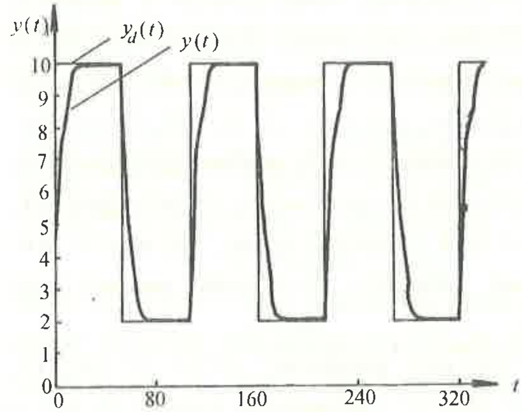

Fig. 4 Output of the plant for Example 2

4.2 Temperature control of the rotary kiln furnace

After system simulation procedure, the proposed FLNC control schems is applied to the temperature control of industrial rotary kiln. The temperature control system can be divided into five main components: the rotary kiln furnace, the temperature sensor, module, the programmable input-output interface board, the microcomputer, and the actuator. The interface circuit board consists of an analogue-to-digital (A/D) convertor, a digital-to-analogue (D/A) convertor and a programmable peripheral interface device. An external clock is designed to operate the A/D and D/A convertor. The microcomputer used in the system is the super-386, with an Intel 80386 32-bit CPU with a 40MHz clock speed. The FLNC control programs are written using Turbo-C to provide the control input to the actuator through the D/A and also to measure the output temperature.

In the experiment, the sampling time in 30 seconds, and the setpoints are 500℃, 1000℃ respectively. Fig.5 illustrates the temperature response of the kiln furnace.

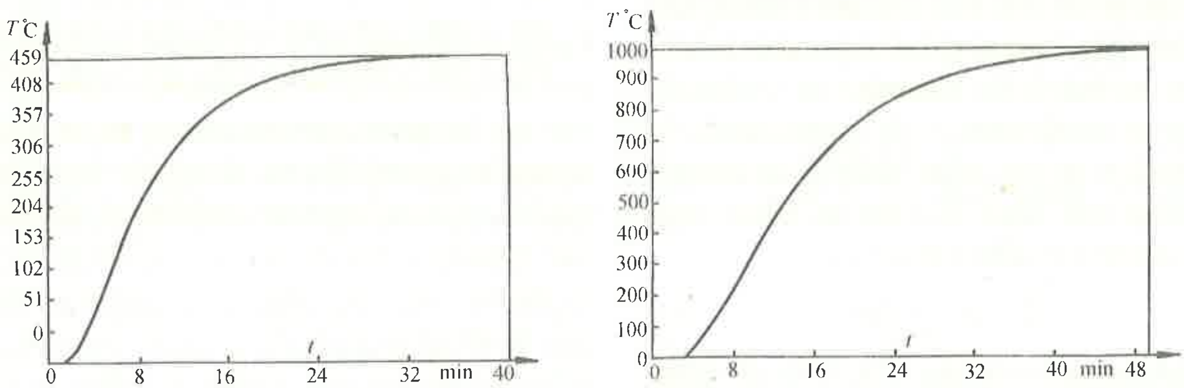

Fig. 5 The temperature response of the kiln furnace

5 Conclusion

A new intelligent control system based on the genetic algorithm and the fuzzy neural networks has been proposed in this paper. The simulation results and the practical application of the industrial rotary kiln furnace show that the proposed control system has two important characteristics: adaptation and learning. It can handle some nonlinear, slow time-varying, and stochastic disturbed process control problem, and can obtain good control performance. The proposed control scheme can also be applied to complex process control.

References

1 Lec Chng-chien. A self-learning rule-based controller employing approximate reasoning and neural net concepts. Int. J. Intelligent Systems, 1991, 16(3): 71 - 93

2 Kosko B. Neural networks and fuzzy system. Prentice Hall, 1992

3 Jang J S. ANFIS: Adaptive-network-based fuzzy inference system. IEEE Trans. System Man and Cybemetics, 1993, 23(3): 665 - 685

4 Goldberg D E. Genetic algorithms in search optimization, and machine learning. Addison-Wesley, Reading, 1989

5 Wang Yaonan. Intelligent control intergrating expert system and neural networks. Advances in Modelling & Analysis France Press, 1994, 43 (4): 23 - 28

6 Wang Yaonan. An adaptive control using fuzzy logic neural network and its application. Control Theroy and Application, 1995, 12(4): 437 - 444

7 Wang Yaonan and Tong T S. Real-time expert intelligent control system REICS. Proc. of IFAC AARTC'92, 1992, 1: 307 - 312

8 Wang Yaoman. Intelligent Control System. Changsha: Hunan University Press, 1996(in Chinese)

本文作者简介

王耀南 1957年生.博士,湖南大学电气工程系教授,博士生导师.主要从事智能信息处理,人工智能,智能机器人与视觉,智能控制理论与应用.

张昌凡 1960年生.副教授,湖南大学电气与信息工程学院博士生.主要从事智能控制理论与应用,变结构控制,交流伺服控制与应用.

(Continued from page 885)

Stability analysis of APRCA2: The intelligent marking capability of APRCA was incorporated into EPRCA and a new scheme called APRCA2 was produced. APRCA2 solves the problem of source-bottleneck. The other parts of APRCA2 is the same as APRCA. So from the proof of instability of APRCA, we can see APRCA2 is unstable too.

5 Conclusion

In the present work, the stability of APRCA and APRCA2 were studied. From the analysis of APRCA and APRCA2, we found they are unstable in some conditions. This is the first paper that has made the discovery and we hope it will give useful advice in practical engineering.

References

1 Sathaye S. ATM Forum traffic management specification version 4.0. ATM Forum: aftm - 0056.000[EB/OL]. http://www.atmforum.com., 1996-04

2 Siu K Y and Tzeng H Y. Adaptive proportional rate controls (APRC) with intelligent congestion indication, ATM Forum Contribution 94-0888 [EB/OL]. http://www.atmforum.com., 1994-09

3 Siu K Y and Tzeng H Y. Limits of performance in rate-based control schemes, ATM Forum Contribution 94-1078[EB/OL]. http://www.atmforum.com., 1994-11

4 Siu K Y and Tzeng H Y. Intelligent congestion control for ABR service in ATM networks, Computer Communication Review, 1995, 24(5):

5 Roberts L. Enhanced PRCA (Proportional Rate-Control Algorithm), ATM Forum Contribution 94-0732R1[EB/OL]. http://www.atmforum.com.

6 Tsai W K, Ge Y and Huang G. Stability analysis of intelligent marking EPRCA for ABR congestion control in ATM. Inforcom'97, Kobe Japan, 1997

7 Ritter M. Network buffer requirements of the rate-based control mechanism for ABR Service. IEEE Infocom'96, San Francisco, USA, 1996

本文作者简介

王锦山 1971年生.1993年和1996年分别获得西安电子科技大学工学学士和工学硕士学位,1999年于华南理工大学电子与通信工程系获工学博士学位.现在广东省邮电科学技术研究院工作.目前研究方向:ATM交换机、高速路由器以及采用ATM与IP技术结合技术的交换机.

韦 岗 1963年生.华南理工大学电子与通信工程系教授,系主任,博士生导师.主要研究方向:通信理论,信号处理.

高敬添 1956年生.香港城市大学电子工程系副教授.主要研究方向:ATM网络理论.