

# Replacing Fuzzy Systems with Neural Networks

Tiantian Xie, Hao Yu, and Bogdan Wilamowski  
 Auburn University, Alabama, USA,  
[txz0004@auburn.edu](mailto:txz0004@auburn.edu), [hzy0004@auburn.edu](mailto:hzy0004@auburn.edu), [wilam@ieee.org](mailto:wilam@ieee.org)

**Abstract.** In this paper, a neural architecture which gives identical TSK fuzzy system is proposed based on the area selection concept in neural network design. Instead of using traditional membership functions for selection the range of operation, the monotonic pair-wire or sigmoidal activation function is used. In the comparison to popular neuro-fuzzy systems [18], the proposed approach does not require signal normalization or division. This neural system does not need training process. All parameters of constructed neural networks are directly derived from specifications of fuzzy systems.

**Keywords:** Fuzzy system, neural networks, Neural-Fuzzy

## I. INTRODUCTION

CONVENTIONAL controllers, such as a PID controller, are broadly used for linear processes [1-3]. In real life, most processes are nonlinear. Nonlinear control [4-6] is considered as one of the most difficult challenges in modern control theory. While linear control system theory has been well developed, it is the nonlinear control problems that cause the most headaches. Traditionally, a nonlinear process has to be linearized first before an automatic controller can be effectively applied [7]. This is typically achieved by adding a reverse nonlinear function to compensate for the nonlinear behavior so the overall process input-output relationship becomes somewhat linear.

The issue becomes more complicated if a nonlinear characteristic of the system changes with time and there is a need for an adaptive change of the nonlinear behavior. These adaptive systems are best handled with methods of computational intelligence such as neural networks and fuzzy systems [8].

In this paper, a neural architecture [9], derived from fuzzy system and neural networks, will be introduced, and compared with classic fuzzy systems and traditional neuro-fuzzy systems [10], based on a surface approximation problem.

The studying case can be described as a nonlinear control surface, shown in Fig. 1. All points (2601 points in Fig. 1a and 36 points in Fig. 1b) in the surface are calculated by the equation.

$$z = (x - 4)^3 + 4(y - 3)^2 \quad (1)$$

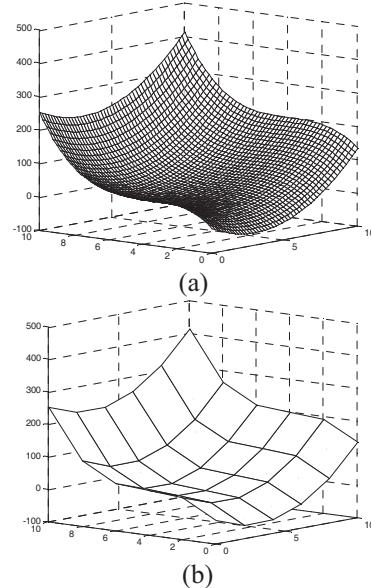


Fig. 1 Required surface obtained from equation (1): (a)  $51 \times 51 = 2601$  points; (b)  $6 \times 6 = 36$  points

## II. FUZZY SYSTEM

The most commonly used architectures for fuzzy system development are the Mamdani fuzzy system [11][12] and TSK (Takagi, Sugeno and Kang) fuzzy system [13][14][15], as shown in Fig. 2. Both of them consist of three blocks: fuzzification block, fuzzy rule block and defuzzification/normalization block. Each of the blocks can be designed differently.

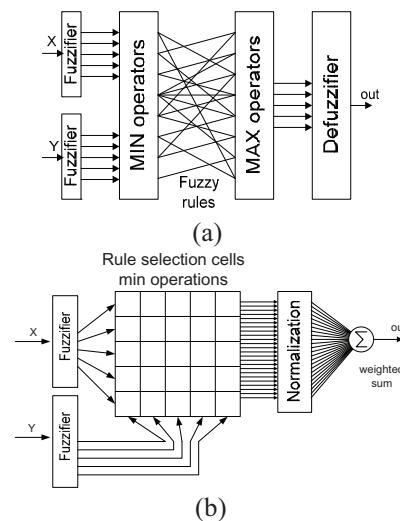


Fig. 2 Block diagram of the two types of fuzzy systems: (a) Mamdani fuzzy system; (b) TSK fuzzy system

### A. Fuzzification

Fuzzification is supposed to convert the analog inputs into sets of fuzzy variables. For each analog input, several fuzzy variables are generated with values between 0 and 1. The number of fuzzy variables depends on the number of member functions in the fuzzification process. Various types of member functions can be used for conversion, such as triangular, trapezoidal or gaussians. One may consider using the combination of them and different types of membership functions result in different accuracies. Fig. 3 shows the surfaces and related accuracies obtained by using the Mamdani fuzzy system with different membership functions, for solving the problem in Fig. 1.

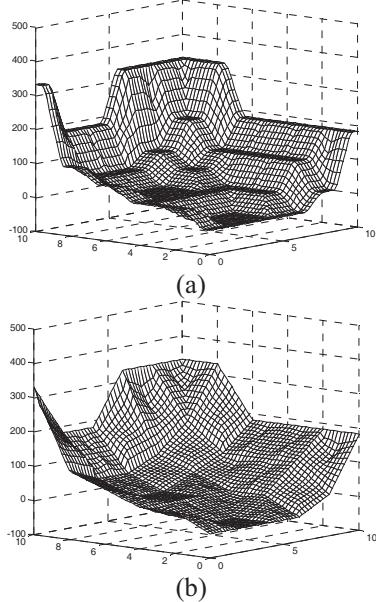


Fig. 3 Control surface using the Mamdani fuzzy systems and 6 membership functions per input: (a) Trapezoidal membership function, averaged sum square error= 657.3997; (b) Triangular membership function, averaged sum square error = 337.3937

One may notice that using the triangular membership functions one can get better surface than from using the trapezoidal membership functions.

The more membership functions are used, the higher accuracy will be obtained. However, very dense functions may lead to frequent controller actions (known as “hunting”), and sometimes this may lead to system instability; on the other hand, more storage is required, because the size of the fuzzy table is increased exponentially to the number of membership functions.

### B. Fuzzy rules

Fuzzy variables are processed by fuzzy logic rules, with MIN and MAX operators. The fuzzy logic can be interpreted as the extended Boolean logic. For binary ‘0’ and ‘1’, the MIN and MAX operators in the fuzzy logic perform the same calculations as the AND and OR operators in Boolean logic, respectively, see Table I; for fuzzy variables, the MIN and MAX operators work as shown in Table II.

TABLE I BINARY OPERATION USING BOOLEAN LOGIC AND FUZZY LOGIC

A	B	A AND B		A OR B	
		MIN(A,B)	MAX(A,B)	MIN(A,B)	MAX(A,B)
0	0	0	0	0	0
0	1	0	0	1	1
1	0	0	0	1	1
1	1	1	1	1	1

TABLE II FUZZY VARIABLES OPERATION USING FUZZY LOGIC

A	B	MIN(A,B)	MAX(A,B)
0.3	0.5	0.3	0.5
0.3	0.7	0.3	0.7
0.6	0.4	0.4	0.6
0.6	0.8	0.6	0.8

### C. Defuzzification

As a result of “MAX or MIN” operations in the Mamdani fuzzy systems, a new set of fuzzy variables is generated, which later have to be converted to an analog output value by defuzzification blocks (Fig. 2a). In the TSK fuzzy systems, the defuzzification block was replaced with normalization and weighted average; MAX operations are not required, instead, a weighted average is applied directly to regions selected by MIN operators.

Fig 4 below shows the result of surfaces using the TSK fuzzy architecture, with different membership functions.

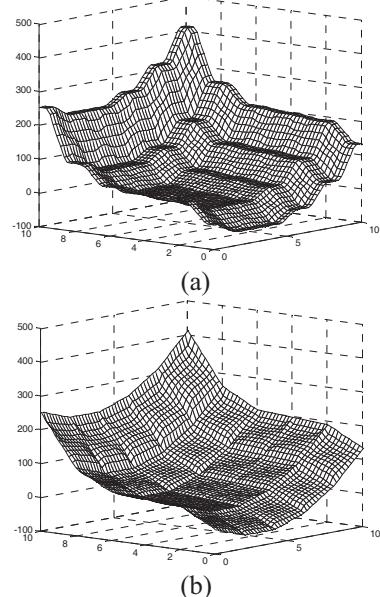


Fig. 4 Control surface using the TSK fuzzy systems and 6 membership functions per input: (a) Trapezoidal membership function, averaged sum square error=214.6959; (b) Triangular membership function, average sum square error=84.2388.

### III. NEURO-FUZZY SYSTEM

Lots of research is devoted to improve the ability of fuzzy systems [16][17], such as evolutionary strategy and neural networks. The combination of fuzzy logic and neural networks is called a neuro-fuzzy system, which is

supposed to result in a hybrid intelligent system by combining human-like reasoning style of neural networks.

#### A. Traditional neural-fuzzy system

Fig. 5 shows the neuro-fuzzy system which attempts to present a fuzzy system in a form of neural network [18].

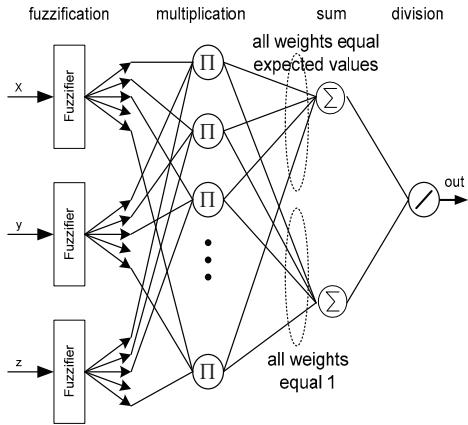


Fig. 5 Neuro-fuzzy system

The neuro-fuzzy system consists of four blocks: fuzzification, multiplication, summation and division. The fuzzification block translates the input analog signals into fuzzy variables by membership functions. Then, instead of MIN operations in classic fuzzy systems, product operations (signals are multiplied) are performed among fuzzy variables. This neuro-fuzzy system with 5product encoding is more difficult to implement but it can generate a slightly smoother control. The summation and division layers perform defuzzification translation. The weights on upper sum unit are designed as the expecting values (both the Mamdani and TSK rules can be used); while the weights on the lower sum unit are all “1”.

Note that, in this type of neuro-fuzzy systems, only the architecture resembles neural networks because cells there perform different functions than neurons, such as signal multiplication or division.

#### B. Proposed Neural System

The structure on Fig 5 actually does not deserve the word “neural” in theory narrative. There is always not much similarity to operation of neurons, which are not capable to perform signal by signal multiplication or division.

In a neural system, a single neuron can divide input space by line, plane, or hyper plane, depending on the problem dimensionality. In order to select just one region in n-dimensional input space, more than  $(n+1)$  neurons are required. For example, to separate a rectangular pattern, 4 neurons are required, as is shown in Fig. 6. If more input clusters should be selected then the number of neurons in the hidden layer should be properly multiplied. If the number of neurons in the hidden layer is not limited, then all classification problems can be solved using the three layer network.

With the concept shown in Fig. 6 fuzzifiers and MIN operators used for region selection can be replaced by a simple neural network architecture. In this example, the

two analog inputs, each with five membership functions, can be organized as a two-dimensional input space which was divided by six neurons horizontally (from line  $a$  to line  $f$ ) and by six neurons vertically (from line  $g$  to line  $l$ ), as shown in Fig. 7. The corresponding neural network is shown in Fig. 8. Neurons in the first layer are corresponding to the lines indexed from  $a$  to  $l$ . Each neuron is connected only to one input. For each neuron input, weight is equal to +1 and the threshold is equal to the value of the crossing point on the  $x$  or  $y$  axis. The type of activation functions of neurons in the first layer decides the type of membership functions of the fuzzy system, as shown in Fig. 9.

Neurons in the second layers are corresponding to the sections indexed from 1 to 25. Each of them has two connections to lower boundary neurons with weights of +1 and two connections to upper boundary neurons with weights of -1. Thresholds for all these neurons in the second layer are set to 3.

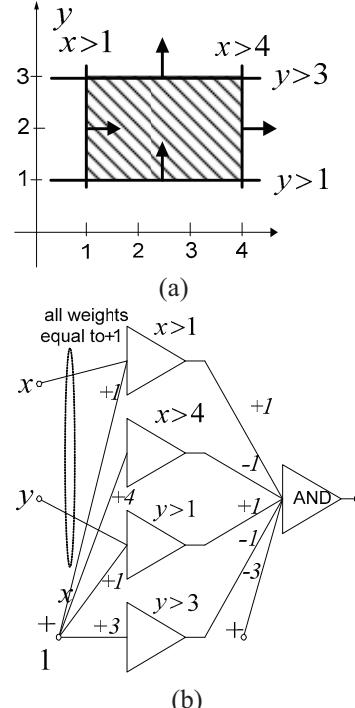


Fig. 6 (a) Separation of the rectangular area on a two dimensional input space; (b) designed neural network to fulfill this task

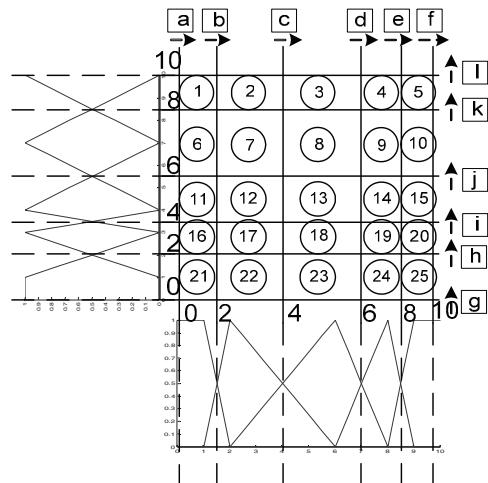


Fig. 7 Two-dimensional input plane separated vertically and horizontally by six neurons in each direction

Weights of the upper sum unit in the third layer have values corresponding to the specified values in selected areas. The specified values can be obtained from either the fuzzy table (by Mamdani rule), or the expected function values (by TSK rule). Weights of the lower sum unit are equal to "1". All neurons in Fig. 8 have a unipolar activation function and if the system is properly designed, then for any input vector in certain areas only the neuron of this area produces +1 while all remaining neurons have zero values. In the case of when the input vector is close to a boundary between two or more regions, then all participating neurons are producing fractional values and the system output is generated as a weighted sum. The fourth layer performs such a calculation: the upper sum divided by the lower sum. Like the neuro-fuzzy system in Fig. 5, the last two layers are used for defuzzification.

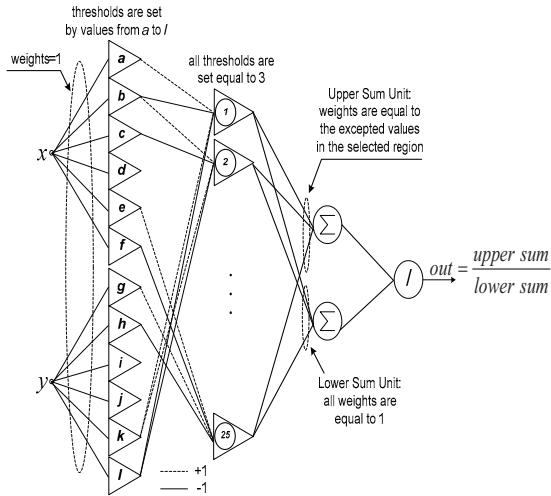


Fig. 8 The neural network performing the function of fuzzy system

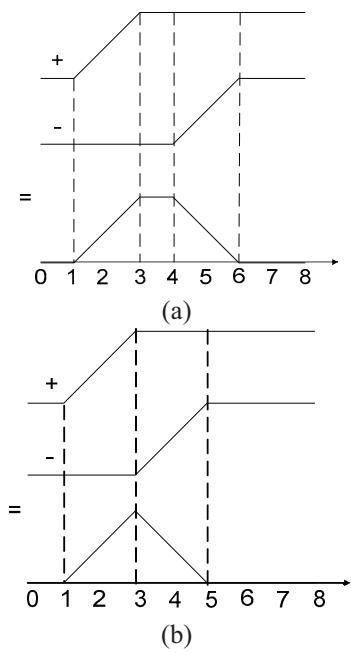


Fig. 9 Construction of membership functions by neurons' activation functions: (a) Trapezoidal membership function; (b) Triangular membership function.

Using this concept of neural system, the result surfaces with different combination of activation functions, can be obtained as shown in Fig. 10.

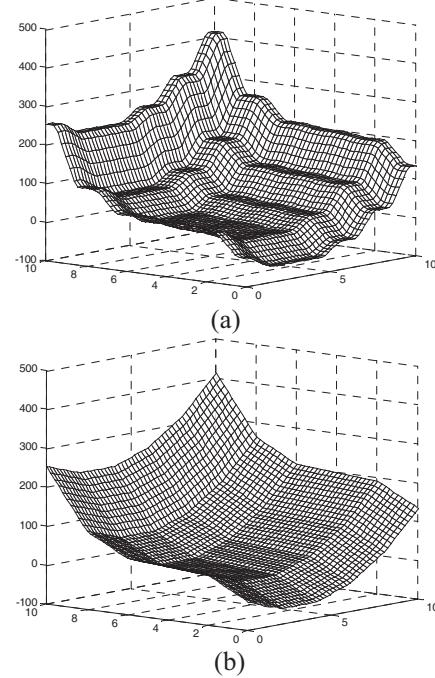


Fig. 10 Control surface using neural system in Fig. 8: (a) using combination of activation functions in Fig. 9a, average sum square error =240.9906; (b) using combination of activation functions in Fig. 9b, average sum square error=60.8369.

Neurons with sigmoidal activation functions can also be used in the proposed neural architecture

$$y = \frac{\rho}{1 + e^{-kx}} \quad (2)$$

where:  $\rho$  and  $k$  are parameters to control the shape of activation functions.

Membership function constructed by sigmoidal activation functions is shown in Fig. 11. The result surfaces with different parameters are obtained as shown in Fig. 12.

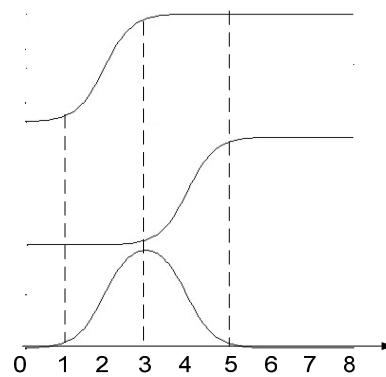


Fig. 11 Construction of membership functions by neurons' sigmoidal activation functions.

## REFERENCES

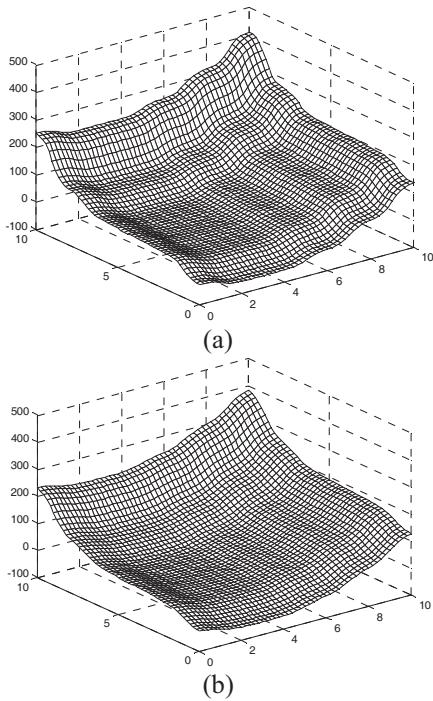


Fig. 12 Control surface using neuro-fuzzy system with sigmoidal function, (a)  $\rho=1$ ,  $k=4$ , average sum square error=141.7895, (b)  $\rho=0.96$ ,  $k=2.7$ , average sum square error=47.334.

From the experimental results, one may notice that, using the proposed neural architecture, the best solution is obtained by using the sigmoidal activation function for each neuron.

## IV. CONCLUSION

The neural architecture, introduced in this paper, improves the performance of classic fuzzy systems. Being different from traditional neuro-fuzzy systems (Fig. 5), the proposed architecture (Fig. 8) is based on neuron design. All parameters of neural networks are directly derived from requirements specified for a fuzzy system and there is no need for a training process.

Both the traditional neuro-fuzzy system and proposed neural architectures got the same errors in the surface approximation problem. However, the proposed system does not use the signal multiplication units as the traditional neuro-fuzzy system in Fig. 5, which simplifies the hardware implementation.

With the properties described in the paper, one may conclude reasonably that the proposed neural system can replace both classic fuzzy systems and the traditional neuro-fuzzy systems.

- [1] Wang Hui Yang Yongbo Liu Meiyu, "Fuzzy-PID control in the Application of Multi-purpose Vehicles of Road Snow Plowing," *International conference on Web Information Systems and Mining, 2009. WISM 2009*, pp. 246-250, Nov. 2009.
- [2] Shenglin Mu Tanaka, K., Yuji Wakasa, Takuya Akashi, Yuki Nishimura, Masato Oka, "Intelligent IMC-PID Control for Ultrasonic Motor," *ICCAS-SICE, 2009*, pp. 1911-1915, Aug. 2009.
- [3] Jingqing Han, "From PID to Active Disturbance Rejection Control," *IEEE Trans. on Industrial Electronics*. vol. 56, no. 3, pp. 900-906, 2009.
- [4] Coutinho, D.F. Da Silva, J.M.G., "Computing estimates of the region of attraction for rational control systems with saturating actuators," *Control Theory & Applications, IET*, vol. 4, no. 3, pp. 315-325, March 2010.
- [5] Irwin, G.W. Chen, J. McKernan, A. Scanlon, W.G., "Co-design of predictive controllers for wireless network control," *Control Theory & Applications, IET*, vol. 4, no. 2, pp. 186-196, Feb. 2010.
- [6] J. A. Farrell, M. M. Polycarpou, "Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches [Book review]," *IEEE Trans. on Neural Networks*, vol. 19, no. 4, pp. 731-732, April, 2008.
- [7] B. M. Wilamowski and J. Binfe, "Microprocessor Implementation of Fuzzy Systems and Neural Networks," *International Joint Conference on Neural Networks (IJCNN'01)*, pp. 234-239, Washington DC, July 15-19, 2001.
- [8] B. M. Wilamowski, "Neural Networks and Fuzzy Systems," chapter 32 in *Mechatronics Handbook* edited by Robert R. Bishop, CRC Press, pp. 33-1 to 32-26, 2002.
- [9] B. M. Wilamowski, R. C. Jaeger, and M. O. Kaynak, "Neuro-Fuzzy Architecture for CMOS Implementation," *IEEE Transaction on Industrial Electronics*, vol. 46, No. 6, pp. 1132-1136, Dec. 1999.
- [10] D. V. Prokhorov, "Intelligent Control Systems Using Computational Intelligence," *IEEE Trans. on Neural Networks*, vol. 18, no. 2, pp. 611-612, Feb. 2007.
- [11] E. H. Mamdani, "Application of Fuzzy Algorithms for Control of Simple Dynamic Plant," *IEEE Proceedings*, Vol. 121, No. 12, pp. 1585-1588, 1974.
- [12] M. McKenna and B. M. Wilamowski, "Implementing a Fuzzy System on a Field Programmable Gate Array," *International Joint Conference on Neural Networks (IJCNN'01)*, pp. 189-194, Washington DC, July 15-19, 2001.
- [13] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Application to Modeling and Control," *IEEE Transactions on System, Man, Cybernetics*, Vol. 15, No. 1, pp. 116-132, 1985.
- [14] Sugeno and G. T. Kang, "Structure Identification of Fuzzy Model," *Fuzzy Sets and Systems*, Vol. 28, No. 1, pp. 15-33, 1988.
- [15] B.M. Wilamowski and J. Binfe, "Do Fuzzy Controllers Have Advantages over Neural Controllers in Microprocessor Implementation," *Proc. of 2-nd International Conference on Recent Advances in Mechatronics - ICRA'99*, Istanbul, Turkey, pp. 342-347, May 24-26, 1999.
- [16] J. J. Cupal and B. M. Wilamowski, "Selection of Fuzzy Rules Using a Genetic Algorithm," *proceedings of Word Congress on Neural Networks*, San Diego, California, USA, vol. 1, pp. 814-819, June 4-9, 1994.
- [17] B. M. Wilamowski and R. C. Jaeger, "Implementation of RBF Type Networks by MLP Networks," *IEEE International Conference on Neural Networks*, Washington, DC, pp. 1670-1675, June 3-6, 1996.
- [18] Masuoka R., Watanabe N., Kawamura A., Owada Y., Asakawa K., "Neurofuzzy system-Fuzzy inference using a structured neural network", *Proceedings of the International Conference on FuzzyLogic&Neural Networks*, Hzuka, Japan, pp.173-177, July20-24,1990.