Neuro-Fuzzy System

Hao Yu Auburn University

Tiantian Xie Auburn University

Bogdan M. Wilamowski Auburn University

()

20.1	Introduction	20- 1
20.2	Fuzzy System	20- 1
	Fuzzification • Fuzzy Rules • Defuzzification	
20.3	Neuro-Fuzzy System	20 -3
	Structure One • Structure Two	
20.4	Conclusion	20- 8
Refer	20 -8	

20.1 Introduction

Conventional controllers, such as PID controller, are broadly used for linear processes. In real life, most processes are nonlinear. Nonlinear control 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 most headaches. Traditionally, a nonlinear process has to be linearized first before an automatic controller can be effectively applied [WB01]. 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 [W02].

In this chapter, neuro-fuzzy system [WJK99], as the combination of fuzzy system and neural networks, will be introduced, and compared with classic fuzzy systems, based on a simple case (Figure 20.1).

The studying case can be described as the nonlinear control surface, shown in Figure 20.1. All points (441 points in Figure 20.1a and 36 points in Figure 20.1b) in the surface are calculated by the equation

$$z = 1.1 \exp\left(-0.07 \left(x - 5\right)^2 - 0.07 \left(y - 5\right)^2\right) - 0.9$$
(20.1)

20.2 Fuzzy System

The most commonly used architectures for fuzzy system development are Mamdani fuzzy system [M74,MW01] and TSK (Takagi, Sugeno, and Kang) fuzzy system [TS85,SK88,WB99], as shown in Figure 20.2. Both of them consist of three blocks: fuzzification, fuzzy rules, and defuzzification/normalization. Each of the blocks could be designed differently.

 (\bullet)

Intelligent Systems



()

FIGURE 20.1 Required surface obtained from Equation 20.1: (a) $21 \times 21 = 441$ points and (b) $6 \times 6 = 36$ points.



FIGURE 20.2 Block diagram of the two types of fuzzy systems: (a) Mamdani fuzzy system and (b) TSK fuzzy system.

20.2.1 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 fuzzification process. Various types of member functions can be used for conversion, such as triangular and trapezoidal. One may consider using the combination of them and different types of membership functions result in different accuracies. Figure 20.3 shows the surfaces and related accuracies obtained by using Mamdani fuzzy system with different membership functions, for solving the problem in Figure 20.1.

One may notice that using triangular membership functions can get better surface than from using 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 leads to system instability; on the other hand, more storage is required, because the size of fuzzy table is increased exponentially to the number of membership functions.

20.2.2 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

20-2

()

۲

Neuro-Fuzzy System



()

FIGURE 20.3 Control surface using Mamdani fuzzy systems and 5 membership functions per input: (a) trapezoidal membership function, error = 3.8723 and (b) triangular membership function, error = 2.4799.

		/ - F			
A	В	A AND B	MIN(A,B)	A OR 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 20.1
 Binary Operation Using Boolean Logic and Fuzzy Logic

in the fuzzy logic perform the same calculations as the AND and OR operators in Boolean logic, respectively, see Table 20.1; for fuzzy variables, the MIN and MAX operators work as shown in Table 20.2.

20.2.3 Defuzzification

As a result of "MAX of MIN" operations in Mamdani fuzzy systems, a new set of fuzzy variables is generated, which later has to be

converted to an analog output value by defuzzification blocks (Figure 20.1a). In 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.

Figure 20.4 shows the result surfaces using TSK fuzzy architecture, with different membership functions.

20.3 Neuro-Fuzzy System

A lot of research is devoted to improve the ability of fuzzy systems [WJ96], such as evolutionary strategy and neural networks [CW94]. The combination of fuzzy logic and neural networks is called neuro-fuzzy system, which is supposed to result in a hybrid intelligent system by combining human-like reasoning style of neural networks.

20.3.1 Structure One

Figure 20.5 shows the neuro-fuzzy system which attempts to present a fuzzy system in a form of neural network [RH99].

()

TABLE 20.2Fuzzy VariablesOperation Using Fuzzy Logic

A	В	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

20-3

۲





()

FIGURE 20.4 Control surface using TSK fuzzy systems and 5 membership functions per input: (a) trapezoidal membership function, error = 2.4423, and (b) triangular membership function, error = 1.5119.



FIGURE 20.5 Neuro-fuzzy system.

The neuro-fuzzy system consists of four blocks: fuzzification, multiplication, summation, and division. 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 product encoding is more difficult to implement [OW96], but it can generate a slightly smoother control surface (see Figures 20.6 and 20.7). The summation and division layers perform defuzzification translation. The weights on upper sum unit are designed as the expecting values (both Mamdani and TSK rules can be used); while the weights on the lower sum unit are all "1."

Figures 20.6 and 20.7 show the surfaces obtained using the neuro-fuzzy system in Figure 20.5, which is smoother than the surfaces in Figures 20.3 and 20.4.

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

()

۲

Neuro-Fuzzy System



FIGURE 20.6 Control surface using neuro-fuzzy system in Figure 20.5, Mamdani rule for weight initialization of the upper sum unit and 5 membership functions per input: (a) trapezoidal membership function, error = 3.8723 and (b) triangular membership function, error = 2.4468.



FIGURE 20.7 Control surface using neuro-fuzzy system in Figure 20.5, TSK rule for weight initialization of the upper sum unit and 5 membership functions per input: (a) trapezoidal membership function, error = 2.4423 and (b) triangular membership function, error = 1.3883.

20.3.2 Structure Two

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, four neurons are required, as is shown in Figure 20.8. 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 on Figure 20.8, fuzzifiers and MIN operators used for region selection can be replaced by 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 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 Figure 20.9. The corresponding neural network is shown in Figure 20.10. 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 Figure 20.11.

20-5

()

()

(



FIGURE 20.8 Separation of the rectangular area on a two dimensional input space and desired neural network to fulfill this task.





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.

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."

20-6

۲

۲



FIGURE 20.10 The neural network performing the function of fuzzy system.



FIGURE 20.11 Construction of membership functions by neurons' activation functions: (a) trapezoidal membership function and (b) triangular membership function.

All neurons in Figure 20.8 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

20-7

۲

۲



()

FIGURE 20.12 Control surface using neuro-fuzzy system in Figure 20.10: (a) using combination of activation functions in Figure 20.11a, error = 2.4423 and (b) using combination of activation functions in Figure 20.11b, error = 1.3883.

divided by the lower sum. Like the neuro-fuzzy system in Figure 20.5, the last two layers are used for defuzzification.

Using this concept of neuro-fuzzy system, the result surfaces with different combination of activation functions, can be obtained as shown in Figure 20.12.

It was shown above that a simple neural network of Figure 20.10 can replace a TSK neuro-fuzzy system in Figure 20.5. All parameters of this network are directly derived from requirements specified for a fuzzy system and there is no need for a training process.

20.4 Conclusion

()

The chapter introduced two types of neuro-fuzzy architectures, in order to improve the performance of classic fuzzy systems. Based on a given example, the classic fuzzy systems and the neuro-fuzzy systems, with different settings, are compared. From the comparison results, the following conclusions can be drawn:

- In the same type of fuzzy system, using triangular membership functions can get better results than those from using the same number of trapezoidal membership functions.
- With the same membership function, TSK (Takagi, Sugeno, and Kang) fuzzy systems perform more accurate calculation than Mamdani fuzzy system.
- The neuro-fuzzy system in Figure 20.5 makes a slightly improvement on the accuracy, with the cost of using signal multiplication units, which are difficult for hardware implementation.
- The neuro-fuzzy system in Figure 20.10 does the same job as the neuro-fuzzy system with TSK rule in Figure 20.5.

References

- [CW94] J. J. Cupal and B. M. Wilamowski, Selection of fuzzy rules using a genetic algorithm, *Proceedings of Word Congress on Neural Networks*, San Diego, CA, June 4–9, 1994, vol. 1, pp. 814–819.
- [M74] E. H. Mamdani, Application of fuzzy algorithms for control of simple dynamic plant, IEEE Proceedings, 121(12), 1585–1588, 1974.
- [MW01] M. McKenna and B. M. Wilamowski, Implementing a fuzzy system on a field programmable gate array, *International Joint Conference on Neural Networks (IJCNN'01)*, Washington DC, July 15–19, 2001, pp. 189–194.

20-8

Journal of Circuits, Systems and Computers, 6(2), 171–184, April 1996. [RH99] D. Rutkowska and Y. Hayashi, Neuro-fuzzy systems approaches, International Journal of Advanced Computational Intelligence, 3(3), 177–185, 1999.

۲

- [SK88] M. Sugeno and G. T. Kang, Structure identification of fuzzy model, *Fuzzy Sets and Systems*, 28(1), 15–33, 1988.
- [TS85] T. Takagi and M. Sugeno, Fuzzy identification of systems and its application to modeling and control, *IEEE Transactions on System, Man, Cybernetics*, 15(1), 116–132, 1985.
- [W02] B. M. Wilamowski, Neural networks and fuzzy systems, Chapter 32 in *Mechatronics Handbook* edited by R. R. Bishop, CRC Press, Boca Raton, FL, 2002, pp. 33-1–32-26.
- [WB01] B. M. Wilamowski and J. Binfet, Microprocessor implementation of fuzzy systems and neural networks, *International Joint Conference on Neural Networks* (*IJCNN'01*), Washington DC, July 15–19, 2001, pp. 234–239.
- [WB99] B. M. Wilamowski and J. Binfet, Do fuzzy controllers have advantages over neural controllers in microprocessor implementation, *Proceedings of the Second International Conference on Recent Advances in Mechatronics—ICRAM'99*, Istanbul, Turkey, May 24–26, 1999, pp. 342–347.
- [WJ96] B. M. Wilamowski and R. C. Jaeger, Implementation of RBF type networks by MLP networks, *IEEE International Conference on Neural Networks*, Washington, DC, June 3–6, 1996, pp. 1670–1675.
- [WJK99] B. M. Wilamowski, R. C. Jaeger, and M. O. Kaynak, Neuro-fuzzy architecture for CMOS implementation, *IEEE Transaction on Industrial Electronics*, 46(6), 1132–1136, December 1999.

۲

()

