The Test Platform Of Vehicle Airbag Controller Design

Yi-Jen Mon

Abstract: The airbag system has become a standard equipment of vehicles recently. The purpose of airbag system is to protect the safeties of driver and passengers. How to improve the ignition timing of algorithm of airbag control at the moment of car accident occurrence is very important. In this paper, the test platform of simulation is established such as to assure the corrections of ignition timings for numerous crash data of car accidents; meanwhile, by this platform the ignition algorithm of airbag control development can be effective and robust.

Index Terms: Vehicle, airbag, ignition algorithm design, RFNN.

1 INTRODUCTION

For briefly, the airbag is composed of the sensor, the inflator, the bag, the vehicle interior and electrical control unit (ECU). The airbag should be ignited or not depends on the impact severity of the vehicle. In the past years, many researches for airbag have been developed [1-3]. Most of these algorithms of ignition for airbag are based on crash data then by intelligent method such as fuzzy or neural networks. The most important problem is how to get these crash data? Because these crash data are recorded by real car crash, this will yield much budget and effort to be paid. In my previous works, the adaptive network based fuzzy inference system (ANFIS) [4] has been shown to be an efficient hybrid learning procedure for learning the membership functions of the fuzzy inference rules. The ANFIS belongs to a supervised neural network which reveals an efficient learning ability. In this paper, a test platform is established to test many different scenarios of crash before the real implementation of car crash test. Besides ANFIS, other intelligent algorithms can be feed into this platform to test their performances. In this paper, the recurrent fuzzy neural network (RFNN) is used [5-7]. This can save much time and money for getting a reliable decision algorithm.

2 TEST PLATFORM ESTABLISHMENT AND RFNN DESIGN

The physical sensor element of airbag system is frontal accelerometer which can generate crash pulse when the car accident is occurred. This pulse data will be transformed into digital signal and this signal can be feed into an electrical control unit (ECU) which is the main decision part of airbag system. Finally, the decision of ignition signal will be generated and connected to the airbag to decide the airbag should be trigger or not. For example, a severe crash must ignite bag but light impact should not ignite the bag to avoid unnecessary injuries of people. The intelligent algorithm of this platform can be formula-based, fuzzy, neural network or other more complex algorithms. In this paper, the recurrent fuzzy neural network is used to demonstrate this platform's performance.

- Yi-Jen Mon, Department of Computer Science and Information Engineering, Taoyuan Innovation Institute of Technology, Chung-Li, Taoyuan, Taiwan, R. O. C. monbuy@tiit.edu.tw

The architecture of RFNN is designed as in Fig. 1. Many different characteristics of crash data can be analyzed in advance then use these characteristics as inputs of RFNN to do decision of ignition for airbag. The purpose of the establishment of the test model is to simulate the impact identification from crash data which physical acceleration characteristics are got from the accelerometer during a real vehicle collision. The design of the RFNN is based on four characteristics. To avoid the time consuming trial and error procedures of deciding the membership functions of every fuzzy input variables. The RFNN is a suitable learning neural network. The principle of RFNN is briefly described as follows [5-7]: Figure 1 shows a four-layer neural network comprising the input (the *i* layer), membership (the *j* layer), rule (the *k* layer), and output (the *o* layer) layers. This network is adopted to implement the proposed RFNN. The recurrent feedback is embedded in the network by adding feedback connections in the second layer of the fuzzy neural network. Since the recurrent neuron has an internal feedback loop, it captures the dynamic mapping network. The signal propagation and the basic function in each layer are introduced as follows:

Layer 1 - Input layer: For every node *i* in this layer, the net input and the net output are represented as

$$net_i^1(N) = x_i^1 \tag{1}$$

$$y_i^1(N) = f_i^1\left(net_i^1(N)\right) = net_i^1(N)\ ,\ i = 1,2 \tag{2}$$

where x_i^1 represents the *i*th input to the node of layer 1 and *N* denotes the number of iterations.

Layer 2 - Membership layer: In this layer, each node performs a membership function. The Gaussian function is adopted as the membership function. For the *j*th node

$$net_j^2(N) = -\frac{\left(x_i^2 + y_j^2(N-1)\theta_{ij}^2 - m_{ij}^2\right)^2}{\left(\sigma_{ij}^2\right)^2} \tag{3}$$

$$y_j^2(N) = f_j^2\left(net_j^2(N)\right) = \exp\left(net_j^2(N)\right)\ j = 1,2,\ldots,m \tag{4}$$

where m_{ij}^2 is the mean, σ_{ij}^2 is the standard deviation and θ_{ij}^2 is the feedback gain of the Gaussian function in the *j*th term of the *i*th input linguistic variable x_i^2 to the node of layer 2, respectively, and *m* is the total number of linguistic variables with respect to the input nodes.

Layer 3 - Rule layer: Each node *k* in this layer is denoted by $\prod$, which multiplies the incoming signal and outputs the product. For the *k*th rule node

$$net_k^3(N) = \prod_j w_{jk}^3 x_j^3 \tag{5}$$

$$y_k^3(N) = f_k^3\left(net_k^3(N)\right) = net_k^3(N)\,,\ k = 1,2,\ldots,n \tag{6}$$

where x_j^3 represents the *j*th input to the node of layer 3, the weights w_{jk}^3 between the membership layer and the rule layer are assumed to be unity.

Layer 4 - Output layer: The single node *o* in this layer is labeled as Σ, which computes the overall output as the summation of all incoming signals

$$net_o^4(N) = \sum_k w_{ko}^4 x_k^4 \tag{7}$$

$$y_o^4(N) = f_o^4\left(net_o^4(N)\right) = net_o^4(N)\,,\ o = 1 \tag{8}$$

where the link weight w_{ko}^4 is the output action strength of the *o*th output associated with the *k*th rule, x_k^4 represents the *k*th input to the node of layer 4, and y_o^4 is the output of the recurrent fuzzy neural network controller.

According to the gradient descent method, the weights in the output layer are updated by the following:

$$\dot{w}_{ko}^4 \equiv -\eta_w \frac{\partial s(t)\dot{s}(t)}{\partial w_{ko}^4}$$

$$= -\eta_w \frac{\partial s(t)\dot{s}(t)}{\partial u_{srn}} \frac{\partial u_{srn}}{\partial w_{ko}^4}$$

$$= -\eta_w \frac{\partial s(t)\dot{s}(t)}{\partial u_{rn}} \frac{\partial u_{rn}}{\partial w_{ko}^4}$$

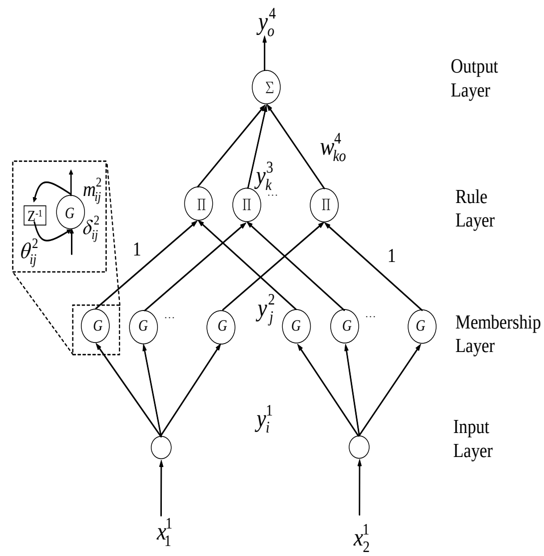

Fig. 1. Network structure of a recurrent fuzzy neural network.

$$= \eta_w k_1 \frac{s(t)}{x(t)} x_k^4 \tag{9}$$

where η_w is the learning rate with a positive constant. Since the weights in the rule layer are unity, only the approximation error term needs to be calculated and propagated by the following:

$$\delta_k^3 \equiv -\frac{\partial s(t)\dot{s}(t)}{\partial u_{srn}} \frac{\partial u_{srn}}{\partial net_o^4} \frac{\partial net_o^4}{\partial y_k^3} \frac{\partial y_k^3}{\partial net_k^3}$$

$$= \frac{s(t)}{x(t)} k_1 w_{ko}^4\,. \tag{10}$$

The multiplication is done in the membership layer and the error term is computed as follows:

$$\delta_j^2 \equiv -\frac{\partial s(t)\dot{s}(t)}{\partial u_{srn}} \frac{\partial u_{srn}}{\partial net_o^4} \frac{\partial net_o^4}{\partial y_k^3} \frac{\partial y_k^3}{\partial net_k^3} \frac{\partial net_k^3}{\partial y_j^2} \frac{\partial y_j^2}{\partial net_j^2} = \sum_k \delta_k^3 y_k^3\,. \tag{11}$$

The update laws of m_{ij}^2, σ_{ij}^2 and θ_{ij}^2 can also be obtained by the gradient search algorithm, i.e.,

$$\dot{m}_{ij}^2 \equiv -\eta_m \frac{\partial s(t)\dot{s}(t)}{\partial m_{ij}^2} \equiv -\eta_m \delta_j^2 \frac{\partial net_j^2}{\partial m_{ij}^2}$$

$$= -\eta_m \delta_j^2 \frac{2\left(x_i^2 + y_j^2(N-1)\theta_{ij}^2 - m_{ij}^2\right)}{\left(\sigma_{ij}^2\right)^2} \tag{12}$$

$$\dot{\sigma}_{ij}^2 \equiv -\eta_\sigma \frac{\partial s(t)\dot{s}(t)}{\partial \sigma_{ij}^2} \equiv -\eta_\sigma \delta_j^2 \frac{\partial net_j^2}{\partial \sigma_{ij}^2}$$

$$= -\eta_\sigma \delta_j^2 \frac{2\left(x_i^2 + y_j^2 (N-1)\theta_{ij}^2 - m_{ij}^2\right)^2}{\left(\sigma_{ij}^2\right)^3} \tag{13}$$

$$\dot{\theta}_{ij}^2 \equiv -\eta_\theta \frac{\partial s(t)\dot{s}(t)}{\partial \theta_{ij}^2} \equiv -\eta_\theta \delta_j^2 \frac{\partial net_j^2}{\partial \theta_{ij}^2}$$

$$= \eta_\theta \delta_j^2 \frac{2\left(x_i^2 + y_j^2 (N-1)\theta_{ij}^2 - m_{ij}^2\right) y_j^2 (N-1)}{\left(\sigma_{ij}^2\right)^2} \tag{14}$$

where η_m , η_σ and η_θ are the learning rates with positive constants.

Based on these membership functions, all the fuzzy rules can be verified by MATLAB fuzzy toolbox. The test platform can be used fly-free to verify different crash data such as shown in Fig. 2. In this platform, there are four diagram windows which can easily check the crash data and verifications of ignitions for airbag. The crash data is shown in the left top diagram and the final decision of ignition is shown in the right bottom diagram. It shows the medium severity ignition signal is generated at about 29 ms after car impact happened.

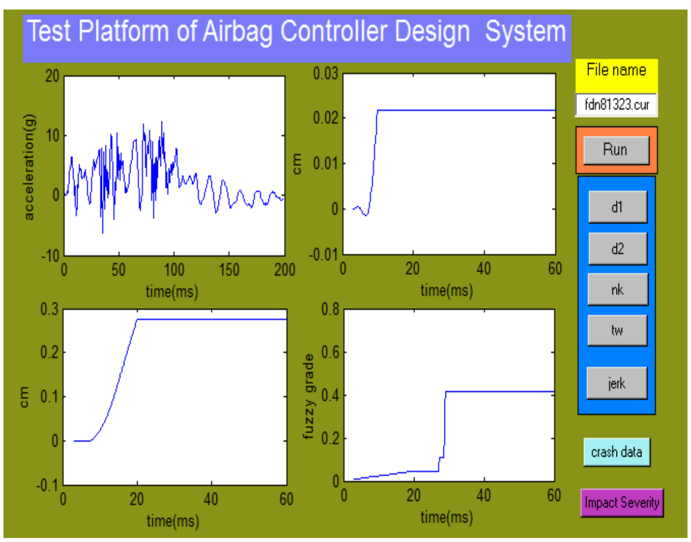

Fig. 2. The test platform of airbag ignition algorithm control system

3 Conclusion

An airbag test platform has been established in this paper. The purpose of airbag is to protect the safeties of driver or passengers. In this paper, the test platform of simulation is established successfully such as to assure the corrections of numerous crash data of car accidents. In implementation this test platform, the algorithm of recurrent fuzzy neural network (RFNN) airbag ignition has been verified and good effectiveness and robustness are also possessed.

Acknowledgment

The author wants to give thanks to the Editor and Reviewers for their valuable comments and partially funded by teacher's research project of Taoyuan Innovation Institute of Technology, Taiwan, R. O. C.

References

[1] Y. J. Mon, K. T. Luo and Y. M. Lin, "Vehicle impact severity identification device," U. S. Patent, US 6,275,755 B1, 2001.

[2] Y. J. Mon, "Airbag controller designed by adaptive-network-based fuzzy inference system (ANFIS)," *Fuzzy Sets and Systems*, Vol. 158, No. 24, pp. 2706-2714, 2007.

[3] S. Altug, M. Y. Chen and H. J. Trussell, "Fuzzy inference systems implemented on neural architectures for motor fault detection and diagnosis," *IEEE Transactions on Industrial Electronics*, Vol. 46, No. 6, pp. 1069-1079, 1999.

[4] J. S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," IEEE Transactions on System, Man and Cybernetics, Vol. 23, No. 3, pp. 665-684, 1993.

[5] F. J. Lin and R. J. Wai, "Hybrid control using recurrent fuzzy neural network for linear-induction motor servo drive," *IEEE Transactions on Fuzzy Systems*, Vol. 9, pp. 102-115, 2001.

[6] Y. J. Mon, C. M. Lin and C. H. Leng, "Recurrent Fuzzy Neural Network Control for MIMO Nonlinear Systems," *Intelligent Automation and Soft Computing*, Vol. 14, No. 4, pp.395-415, 2008.

[7] Y. J. Mon and C. M. Lin, "Supervisory Recurrent Fuzzy Neural Network Control for Vehicle Collision Avoidance System Design," *Neural Computing & Applications*, Vol. 21, No. 8, pp. 2163-2169, 2012.