TRAINABLE FUZZY CONTROLLER

BOGDAN WILAMOWSKI AND RICHARD SANDIGE
Department of Electrical Engineering
University of Wyoming

ABSTRACT:

Two approaches are investigated for determining fuzzy rules by automating the process of rule finding. The first approach uses a local error criterion that provided adequate results but is not a general solution. The second approach using a global error criterion that provided optimal results but required excessive computer time. A combination of both approaches outperform solutions provided by an expert. Both approaches use the classical example of the backing truck to ramp problem.

INTRODUCTION

Neural networks are easy to train, but a fuzzy system requires a sophisticated design process. This process requires expert knowledge to determine the fuzzy rules, and expert intuition is required to define the membership function parameters. On the other hand, output parameter value classification is unknown for neural networks making it difficult to trace the decision process. In contrast, the decision process for fuzzy systems are easily developed.

In this paper an automated synthesis process for fuzzy systems is presented. A fuzzy system like a neural system, has to perform nonlinear mapping of a set of input variables to a set of output variables. Most current research effort is concentrated on the synthesis of fuzzy systems for given input-output relations (Bezdek, 1993; Sugeno and Yasukawa, 1993). Fuzzy systems can be trained in a way similar to neural networks that use back propagation (Horikawa, Furuhashi, and Usikawa, 1992; Hertz and Hu, 1992; Jang, 1992) or more advanced (Wang and Mendel, 1992) algorithms for training. For that purpose special neural structures are usually developed which emulate fuzzy system structures. If input-output relations are known then all fuzzy rules are explicit and the synthesis process is reduced to finding shapes (break points) of membership functions for each input and output variable.

In the case of the backing truck to ramp example (Kong and Kosko, 1992; Kosko, 1992) utilized in this paper, there is no expert to specify the optimum position of the steering angle for an arbitrary truck location. Furthermore a controlled object has certain "inertia", so the correctness of the assumed control variables are not known for some time after the truck drives out of the parking lot. The truck may wreck during the process of backing up if the wrong rules are applied, and it is not always known which rules are wrong.

This paper addresses the problem of automated synthesis of fuzzy rules, when the required control surfaces (input-output relationships) for the controller are not known. It is assumed that a set of membership functions for each variable is given and the model for the controlled object is known. The

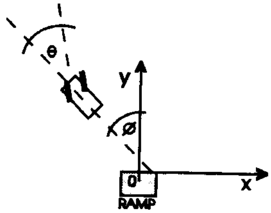


Figure 1. Variables used in the backing truck to ramp problem.

moving truck shown in Figure 1 is described by the following set of equations:

$$\begin{aligned} x_{i+1} &= x_i - r\sin(\phi_i) \\ y_{i+1} &= y_i + r\cos(\phi_i) \\ \phi_{i+1} &= \phi_i + \Theta_i \end{aligned} \qquad (1)$$

where x, y, and ϕ represent the truck's position, while Θ and r represent the steering angle and the incremental driving distance respectively.

Using the same set of membership functions as assumed by Kosko shown in Figure 2, 35 fuzzy rules (5x7) have to be specified. Each fuzzy rule has seven possible output values. Therefore, the process of synthesis requires finding one set of rules out of $7^{35} = 3.79\text{x}10^{29}$ possible combinations. Verifying the correctness of a random set of rules by controlling a model that drives a truck required an excessive amount of computing time on a 486 PC. Therefore, a trial and error method takes too long to find an optimal set of rules.

Two different approaches are investigated for automated rule finding. The first approach assumes an error function is known for each truck location. The second approach uses progressive improvement of fuzzy rules by random fluctuations of the rules around a set of initial values. The second approach is relatively slow and should be used only for final tuning of the fuzzy rules.

APPROXIMATE FUZZY RULE FINDING USING AN ERROR FUNCTION

For the example being considered, the following error function can be specified:

$$E = w_x x^2 + w_y y^2 + w_\phi \phi^2 \qquad (2)$$

where x, y and ϕ are coordinates describing the truck's position and w_x, w_y and w_ϕ are weights assigned for each variable.

Input variable y in equation (2) may be ignored. Instead of driving the truck to the ramp the solution can be found by directing the truck to the target position toward the ramp. When the truck is directed to the target position x = 0 and ϕ = 0 then it is only a matter of time before it will reach the ramp (y = 0). The the error function reduces to the following form:

$$E = w_x x^2 + w_\phi \phi^2 \qquad (3)$$

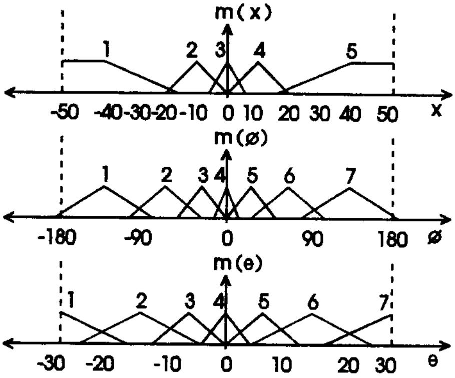


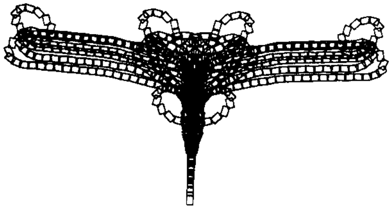
Figure 2. Fuzzy membership functions used in experiments.

The same approach was used by Kosko. For training purposes, it is logical to choose values for the input variables that correspond to the center values of the membership functions as shown in Figure 2. Using x and ϕ, the training set consists of 35 possible starting points. One starting point is chosen for each of the 35 fuzzy rules such that only one specified rule is applicable, and the object is to determine that rule. At each starting point the truck is driven using all seven possible rules. The rule that provides the smallest error function (3) is chosen. To find the error deviation for each rule, one or two iterations are usually enough. Larger step numbers must not be used because the truck may drive out to the position where other fuzzy rules are applicable.

Figure 3a shows the fuzzy rules for the method described above with $w_x = 1$ and $w_\phi = 0.001$. Numbers are used for rules instead of two letter code words where: NB→1, NM→2, NS→3, ZE→4, PS→5, PM→6, and PB→7. Figure 3b shows the traces of the truck driven from the same 35 starting locations using

		x				
		1	2	3	4	5
	1	7	7	7	1	1
	2	2	4	7	7	7
	3	1	1	6	7	7
ϕ	4	1	1	6	7	7
	5	1	1	2	7	7
	6	1	1	1	4	6
	7	7	7	1	1	1

(a)

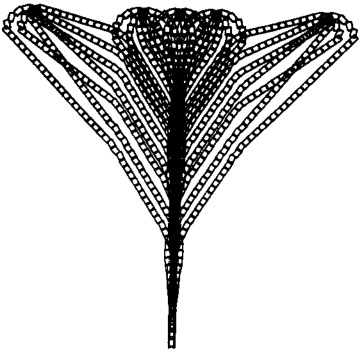


(b)

Figure 3. (a) Fuzzy rules obtained using a local error function with $w_x = 1$ and $w_\phi = 0.001$. (b) Traces of the truck.

				x		
		1	2	3	4	5
	1	7	7	7	7	7
	2	4	6	7	7	7
	3	1	3	5	6	7
ϕ	4	1	2	4	6	7
	5	1	2	3	5	7
	6	1	1	1	2	4
	7	1	1	1	1	1

(a)



(b)

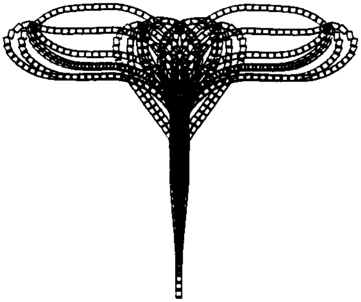
Figure 4. (a) Fuzzy rules obtained using a local error function with $w_x = 1$ and $w_\phi = 0.007$. (b) Traces of the truck.

the fuzzy rules presented in Figures 3a. Figure 4a shows the fuzzy rules with $w_x = 1$ and $w_\phi = 0.007$ while Figure 4b shows the corresponding traces of the truck. The rules specified by an expert (Kong and Kosko, 1992; Kosko, 1992) are shown in Figure 5a and the corresponding truck traces in Figure 5b.

The method described above is very fast but does not provide a general solution. Defining an error function that will decrease monotonically toward one global minimum is not always possible. In many cases an error function may not be smooth and the system gets trapped in local minims as illustrated in Figure 3b. A set of rules obtained using this method strongly depends on the error function. The fuzzy rules obtained using this approach may be correct, but they are not optimal. The results are usually worse than those obtained with fuzzy rules designed by an expert.

			x			
		1	2	3	4	5
	1	5	6	6	7	7
	2	3	5	6	7	7
	3	2	3	5	6	7
ϕ	4	2	3	4	5	6
	5	1	2	3	5	6
	6	1	1	2	3	5
	7	1	1	2	2	3

(a)



(b)

Figure 5. (a) Fuzzy rules specified by an expert. (b) Traces of the truck.

PROGRESSIVE IMPROVEMENT OF FUZZY RULES

In the second method a set of starting positions are chosen. In these experiments we are using the same 35 starting points as described earlier. Instead of using a local error function as given by equations (2) and (3) a global error function is introduced. The global error function represents the total time required by all 35 runs to reach the target position of x = 0 and $\phi = 0$. For constant driving speed, this error function corresponds to the total distance driven by all 35 sample runs before reaching the target position. With an error function that has seven levels of fuzzy rules, the fuzzy rules numbers are randomly changed by ±1. During each try only one fuzzy rule number is changed. If a decrease in the global error function occurs the new and better rule is then used for further experimentation. A similar technique known as "simulated annealing" is used to train the neural network. During the annealing process random fluctuations of the parameters usually change only slightly as the parameters get closer to the solution causing the annealing technique to be very slow and inefficient. Because of the discrete nature of fuzzy rules the magnitude of each fluctuation is not allowed to drop below ±1, therefore convergency is reached very fast. A solution is reached when smaller global error changes stop occurring.

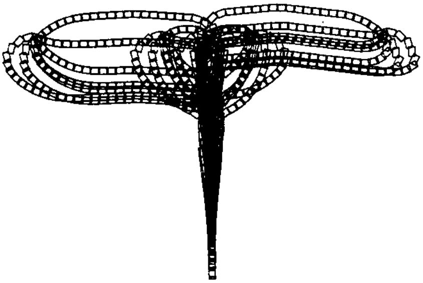
This approach always converges providing very good solutions, but each solution is slightly different. For example, Figures 6a and 6b and Figures 7a and 7b show two different sets of fuzzy rules and their corresponding truck traces. Note that each solution obtained with the algorithm is better than the results obtained in Figure 5b with the set of fuzzy rules shown in Figure 5a that are designed by an expert.

CONCLUSION

Two approaches for the automated finding of fuzzy logic rules were investigated. The first method uses the criterion of minimizing a local error. This method is very fast, but the results obtained are relatively poor. For an

				x		
		1	2	3	4	5
	1	5	6	6	7	7
	2	3	3	7	7	7
	3	2	3	5	7	6
ϕ	4	3	3	4	5	7
	5	1	1	3	5	7
	6	1	1	1	5	6
	7	1	1	1	2	3

(a)

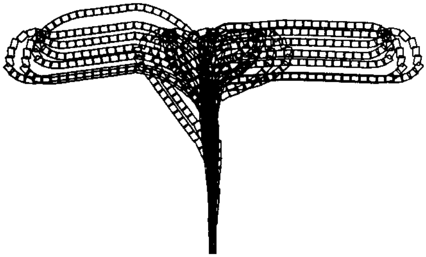


(b)

Figure 6. (a) Fuzzy rules obtained using a global error function. (b) Traces of the truck.

		x				
		1	2	3	4	5
	1	5	7	7	7	7
	2	2	5	7	7	7
	3	1	3	5	7	6
ϕ	4	2	3	4	5	6
	5	2	1	3	6	7
	6	1	1	1	5	6
	7	1	1	1	2	2

(a)



(b)

Figure 7. (a) Fuzzy rules obtained using a global error function. (b) Traces of the truck.

improper local error formula convergency was not assured. The second method uses the criterion of minimizing a global error. This method usually leads to a very good solution, but it was very time consuming. A combination of both methods provides the best solution. The first method was used for finding an approximate solution, and the second method was used for finding a final optimum solution by fine tuning the fuzzy rules.

REFERENCES

Bezdek, J., (1993). "Fuzzy models - What are they, and Why," *IEEE Trans. on Fuzzy Systems*, (1), Feb, 1-6.

Hertz, D. B., Hu, Q., (1992). "Fuzzy-neuro controller for backpropagation networks," *Proceedings of WNN 92*, Feb, 474-478.

Horikawa, S., Furuhashi, T., and Usikawa, Y., (1992). "On fuzzy modeling using fuzzy neural network with the back-propagation algorithm," *IEEE Trans. on Neural Networks*, (3), Sept, 801-806.

Jang, J. R., (1992). "Self-learning fuzzy controllers based on temporal back propagation," *IEEE Trans. on Neural Networks*, (3), Sept, 714-723.

Kong, S., Kosko, B., (1992). "Adaptive fuzzy system for backing up a truck-and-trailer," *IEEE Trans. on Neural Networks*, (3), March, 211-223.

Kosko, B., (1992). *Neural Networks and Fuzzy Systems - A dynamical Systems Approach to Machine Intelligence*, Prentice Hall.

Sugeno, M., Yasukawa, T., (1993). "A fuzzy-logic-based approach to qualitative modeling," *IEEE Trans. on Fuzzy Systems*, (1), Feb, 7-31.

Wang, L. X., Mendel, J. M., (1992). "Fuzzy basis function, universal approximation, and orthogonal least-squares learning," *IEEE Trans. on Neural Networks*, (3), Sept, 807-814.