FUZZY EXPERT SYSTEMS VS. NEURAL NETWORKS — TRUCK BACKER–UPPER CONTROL REVISITED

P.A. Ramamoorthy and Song Huang

Department of Electrical & Computer Engineering,
University of Cincinnati, M.L. #30
Cincinnati, Ohio 45221-0030
FAX: 513–556–7326, Email: pramamoo@nest.ece.uc.edu

ABSTRACT

Research on neural networks and fuzzy logic have progressed on two independent paths. In general, fuzzy logic uses verbal information for handling higher-order logical relations between inputs and outputs which are not crisply defined. On the other hand, neural networks are used to obtain information about systems from large input/output observations and training or learning procedures. From these definitions, it appears that fuzzy logic and neural network fulfill two complementary functions. Hence, a merger of these two concepts could lead to powerful yet flexible knowledge processing tools. This paper provides some insights along these lines using the truck–backer–upper control problem. New network architectures by merging these two concepts and simulation results for the truck–back–upper problem using the new architecture are also shown in this paper.

INTRODUCTION

Both neural network and fuzzy expert system are systems that map an input u (a vector of size $N \times 1$) into an output y (a vector of $M \times 1$) by the function $f: u \rightarrow y$. In a simple neural network, the mapping is performed in the system by weighing each and every inputs, summing the results, subtracting a bias value and passing the result through a non–linear function which may produce a binary or bipolar or continuous value [1,2]. Such networks may be cascaded to propagate the intermediate results to higher levels for more sophisticated problems. In the case of fuzzy expert systems, the ranges of the inputs and outputs are split into smaller and overlapping ranges or fuzzy sets. A fuzzy membership function is associated to each fuzzy subset. The mechanism governing the mapping from the input fuzzy sets to the output fuzzy sets is a collection of fuzzy rules — fuzzy rule base or fuzzy associative memories (FAM) [3,4]. The mapping from the inputs u to the output y is achieved through these fuzzy rules, the membership functions, and defuzzification procedure.

There are similarities and differences between these two mapping systems. The similarities include provision for dealing with imprecise data or data corrupted by noise, having similar primitives or building blocks to produce nonlinear mapping (membership functions, fuzzy rules, MAX–MIN or centroid operations, vs. sigmoid functions in neural networks). The major difference is that fuzzy expert systems use logic rules for inferencing while neural networks are data–driven. Therefore, fuzzy expert systems can be considered as a macroscopic tool for information processing, whereas neural networks are microscopic in nature. The advantage of neural network is their ability to learn the mapping through training. The advantages of fuzzy expert systems are their ability to provide nonlinear mapping through the membership functions and fuzzy rules, and the ability to deal with fuzzy information and incomplete and/or imprecise data. By merging the advantages of these two systems, one can arrive at a more powerful yet more flexible system for inferencing and learning. This concept will be explained through the use of results for the truck–backer–upper control problem.

PROBLEM DEFINITION

The truck backer-upper control is a typical nonlinear control problem where a controller to successfully back up a truck to a loading dock from any reasonable initial location has to be designed. Nguyen and Widrow [5] showed that a nonlinear controller using a two layer neural network architecture with 26 adaptive neural elements can be successfully trained. Recently, Kong and Kosko [6] compared the performance of such a neural network based controller with that of a controller based on fuzzy expert system composed of 35 rules. They observed that even that simple fuzzy expert system lead to smoother trajectories than that produced by the two-layer neural network. If their observations are valid in general, it is desirable to arrive at a logical explanation for the differences in the performances. More importantly, as stated earlier, approaches that can retain the attractive properties of neural networks and at the same time obtain performances comparable to that of fuzzy expert systems need to be developed.

Figure 1 shows the loading zone of the truck–backer–upper problem and inputs and output variables of the system. If enough clearance is given between the truck and the loading dock, then the y-position can be omitted as a input to tune the controller. The ranges of the inputs, x-position and the truck orientation angle θ, and the output, steering signal ϕ, are given as:

$$\phi\text{: } [\ 0, 360];\ \text{x: } [\ 0, 100];\ \theta\text{: } [\ \text{-}30, 30]$$

Having identified the variables and their ranges, fuzzy sub-

CH3051-0/91/0000-0221 $1.00 © 1991 IEEE

sets of the variables must be specified. Fuzzy subsets are simply linguistic terms and their numerical values corresponding to the input and output parameters. They are used to split the total range of the variables into smaller and overlapped ranges. The next step is the selection or identification of the membership functions associated with the various fuzzy subsets. The membership functions simply describe the degree of association of a particular input or output value to the fuzzy subsets belonging to that input or output parameter. The membership functions of the variables for the truck-backer-upper problem are given in Figure 2. Membership functions can have different shapes depending on the designer's preference or knowledge or experience. Triangular and trapezoidal shapes are used here for simpler calculation and better description of the problem.

A fuzzy rule-base or fuzzy associative memories (FAMs) is a collection of fuzzy rules which define or describe the relationship between input fuzzy sets and output fuzzy sets. The rules are in the conventional IF-THEN form, with an antecedent part to describe the conditions and a consequent part to state the conclusions or actions. Figure 3 is the rule-base for the truck- backer-upper problem defined by Kong and Kosko. This rule-base simply indicates the possible ranges for the output values given the input values. The final crisp output values of a fuzzy expert system is determined using a procedure known as defuzzification process. Centroid method is mostly used as the defuzzification procedure and for the truck-backer-upper control problem as well. The trajectory of the truck from a given position backing up to the loading zone controlled by a fuzzy expert system is given in Figure 4. The trajectory of a truck produced by a two layer neural network accordingly is reproduced as Figure 5 (Kong and Kosko). The controller has 24 hidden neurons and trained by over three thousand samples.

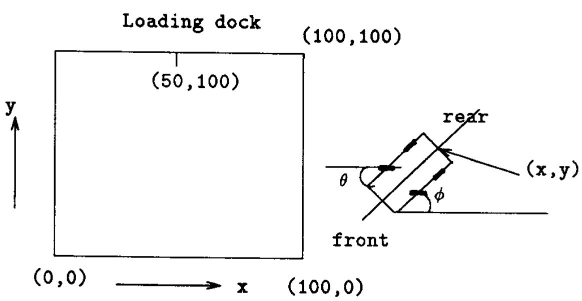

Figure 1. The loading zone of the TBU problem.

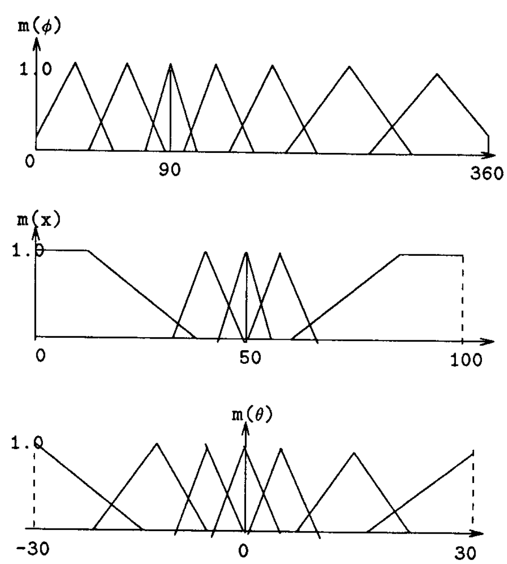

Figure 2. The membership functions of the TBU problem.

angle (φ)	x-position LE	LC	CE	RC	RI
RB	PB	PB	PM	NB	NB
RU	ZE	PS	PM	PB	PB
RV	NB	NM	PS	PM	PB
VE	NB	NM	ZE	PM	PB
LV	NB	NM	NS	PM	PB
LU	NB	NB	NM	NS	ZE
LB	PB	PB	NM	NB	NB

Figure 3. The rulebase of the TBU problem.

The conclusion drawn by Kong and Kosko is that fuzzy expert system controller is superior compared to that of a neural network. If both fuzzy expert systems and neural networks provide robust nonlinear mapping, the difference in the results of these two systems should not be significant. One may argue that the differences may be due to the size of the network, input/output samples used for training and the number of iterations. (But this kind of practical constraints are bound to exist). On the other hand, one can argue that even if such parameters (as the number of nodes) are increased to the maximum practical limit, the neural network performance may not be comparable simply because the fuzzy expert systems use more information in a structured way. We take the later attitude and proceed from these to arrive at a methodology that will combine the best of both worlds, trainability in neural networks and better mapping property of fuzzy expert systems. This will become clear in the next section.

NETWORK REPRESENTATION OF FUZZY EXPERT SYSTEM

Let us examine the steps in implementing a fuzzy expert system. Given the exact input values, we first determine the fuzzy sets to which these inputs belong. If we use a unweighted binary representation for the inputs (i.e. Q lines carrying 0 or 1 for Q quantization levels) and one bit per fuzzy set to indicate if a particular input fallen under that fuzzy set (we call this as input fuzzy set pointer), this step can be represented as an OR network as shown in Figure 6a. The inferencing from the rule-base can be thought of as turning "on or off" of the bits denoting output fuzzy sets based on which input fuzzy set have been selected. Thus, the inferencing can be represented by a two-layer AND-OR (or sum of product) network (the second block in Figure 6a). The defuzzification process can be represented as yet another network whose inputs are the output fuzzy set pointers and the membership function values for a given input value. The outputs of this network will be the final outputs of the fuzzy expert system. This network can be subdirected into smaller networks, each of which can be a 2 or 3 layer perceptron networks as in Figure 6b.

From the above discussion, it is obvious that these three blocks constitute a network representation of a fuzzy expert system. Thus, it is obvious that the performance will be poor (degraded) if we try to represent the tasks of these three individual blocks in a single 2 or 3 layer neural network. One may argue that a 3 layer network is sufficient to represent any nonlinear mapping. But this is only true from a theoretical point of view but does not hold from an engineering point of view.

Having identified the network implementation of a fuzzy expert system, we can use this structure and any additional knowledge (besides large input/output samples) that we may have to find the actual interconnections/weights of the individual blocks. For example, if we assume that the fuzzy sets (of inputs and outputs), the corresponding membership functions and the fuzzy rules are known but the defuzzification procedure is unknown, the architecture of the first two blocks are easy to arrive at and the third block can be trained using the output fuzzy set pointers, membership function values and the desired outputs. Such a training can be very fast as we have represented the third block as a number of smaller sub-blocks. Many variations of these procedure are possible depending upon what kind of information is available. We will be discussing all such possibilities in another paper.

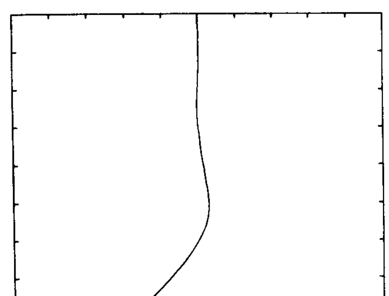

Figure 4. One trajectory of the truck by a fuzzy controller.

Let us now show results based on the above approach. There is no training for the first two blocks of Figure 6a as indicated before. The third block is trained by a) using

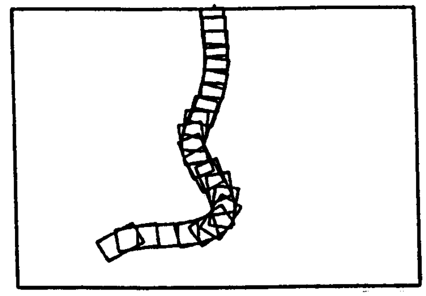

Figure 5. One trajectory of the truck by a NN controller.

both input and output fuzzy set pointers, input x and ϕ values and output values and b) using output fuzzy pointers, input membership function values and input, output values. All these data were generated using the fuzzy controller as explained bellow.

There are no training involved in the first two blocks (Figure 6a). It can be connected by using the information about the input variables and the rule base. The output of this network are directions of output fuzzy set corresponding to input samples, or output fuzzy set pointers. The second network is the controller which uses input variables (x-position and ϕ) to find the θ signal which is fed into a truck emulator to calculate the new x- position, y-position and new ϕ. The inputs of this network also include some fuzzy expert system information, such as membership function values for each input value, or the input fuzzy set pointers which point

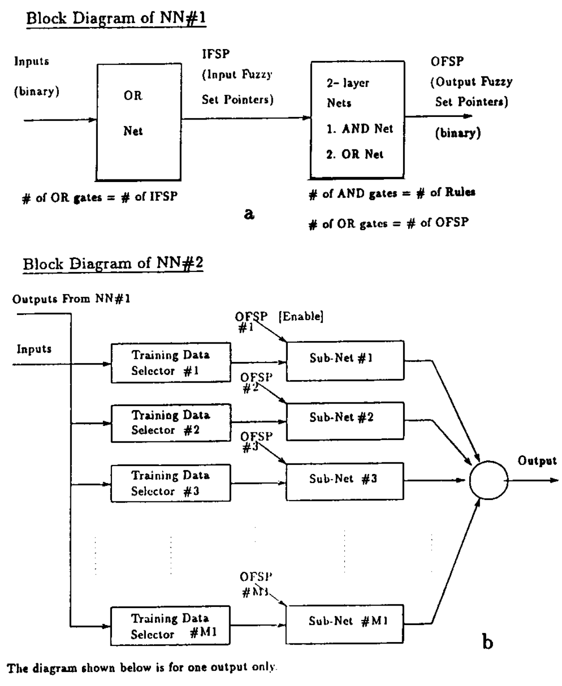

Figure 6. Block diagrams of the FES-NN system.

to the proper fuzzy set for each input. If the former information is used, then the membership function values of each training sample must be known. Whereas only the ranges of the input fuzzy set are needed for the later case.

The truck–backer–upper fuzzy expert system controller has 7 input fuzzy sets for x–position, 5 sets for orientation angle ϕ, and 7 output fuzzy sets for the steering signal θ. Since more accurate results are demanded when the truck is in the center area or near center area, we selected more samples for x–position around 50, and less samples to the extremes. The training samples of ϕ are chosen in the same fashion. It led to 34 x–positions and 72 ϕ angles. Thus 2448 samples are used to train the controller. The y–positions are not used in training, thus simplifying the training process. There are 7 sub–networks in the system. The whole set of training samples are divided into 7 smaller groups according to their belongings to the output fuzzy sets. The largest group contained 826 training samples and the smallest one has 271 samples. Some samples are used more than one subnets due to the overlapping of the fuzzy sets. this brought the total training samples for all subnets to 3624. There are 10 second–layer neurons for every subnet. The backpropagation algorithm was used for the training. The number of iterations for training varies from few hundred (for smaller sample groups) to few thousand (for larger groups). The average square errors are from 0.0005 (for the centered or near center sets) to 0.0015 (for the extreme sets). The training samples were normalized to the range of -0.5 to 0.5. As stated before, either membership function values or input fuzzy set pointers are used with input variables for training the nets. To show the robustness of this architecture, we did the training by using these two different sets information. One truck trajectory produced by using the neural network corresponding to case a) is shown in Figure 7, and Figure 8 shows one trajectory corresponding to case b). It can be noted that both methods produces superior results than the one generated by a two layer neural network as shown by Kong and Kosko for this particular initial condition. Further, our own efforts to train a two layer network with 20 hidden nodes with the same input/output data was only marginally successful and the performance of that controller was very poor.

CONCLUSIONS

A fuzzy expert system is characterized by six attributes: 1) fuzzy input variables, 2) fuzzy output variables, 3) fuzzy sets of the input and output variables, 4) membership functions corresponding to the fuzzy sets, 5) fuzzy rules connecting the input fuzzy sets and output fuzzy sets, and 6) methodology for defuzzification of the fuzzy output. In all real world problems, we will have (or we can infer) useful information about the first four attributes based on the problem at hand. Of these four attributes, only two (number 1 and 2) are presently used in the design of neural network architectures. We showed that by using the other two attributes, one can arrive at new neural network architectures that will provide superior performance and also smaller networks that can be trained rather easily. This concept is proved through the use of the design of a new controller for the truck–backer–upper control problem.

In this work, we considered the training of only one of the three blocks of our new network. However, if information about fuzzy subsets and the rules are not available, one can train the other two blocks as well using some initial knowledge about the input and output variables. Once these blocks are trained, we can sort of pull out the fuzzy rules for further examination and modification. Work along these lines is being carried out presently.

ACKNOWLEDGEMENT

This research work has been supported by a grant NAG #3960 from NASA Lewis Research Center, by UC–NASA Center Grant and a grant N 00014-89-J-1633 from Office of Naval Research.

REFERENCES

1. D.E. Rumelhart and J.L. McClelland, *Parallel Distributed Processing — Explorations in the Microstructure of Cognition*, Cambridge, MA. MIT Press, 1986.
2. R.P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Magazine, pp. 4-22, April 1987.*
3. L.A. Zadeh, "Fuzzy Logic," *IEEE Computer*, pp.83-93, April 1988.
4. B. Kosko, "Fuzzy associative memories," in *Fuzzy Expert Systems* (A. Kandel, ed.), New York: Addison-Wesley, 1987.
5. D. Nguyen and B. Widrow, "The truck backer-upper: an example of self-learning in neural networks," *Preceedings of IJCNN 1989,* vol.2, pp. 357-363, June 1989.
6. S. Kong and B.Kosko, "Comparision of fuzzy and neural truck backer-upper control systems," *Preceedings of IJCNN 1990*, vol.3, pp. 349-358, June 1990.

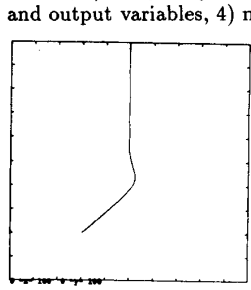

Figure 7. One trajectory by the fuzzy–NN controller.

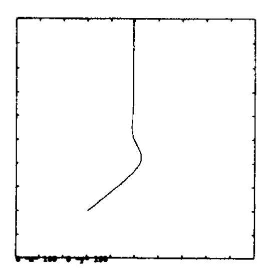

Figure 8. One trajectory by the fuzzy–NN controller.