Fifth IFSA World Congress (1993), 989-992

Architecture for Complex Inference Method

M.H. Lim
J.Y. Leong

School of Electrical & Electronic Engineering
Nanyang Technological University
Singapore 2263
Tel: (65) 799-5408
E-mail: emhlim@ntuvax.ntu.ac.sg

Abstract

In this paper, we describe hardware architecture of fuzzy processors for reasoning involving fuzzy control "heuristics". This we believe will lead to fuzzy systems that are closer to the way humans process domain knowledge for decision making. One noticeable beneficial effect based on our notion of fuzzy heuristics is the significantly reduced number of rules required.

1. Introduction

Very often, humans make use of heuristic rules to accomplish tasks with seemingly little effort. To express such rules, it is more convenient and intuitive to make use of vague linguistic terms. These linguistic terms can be conveniently represented as fuzzy sets. Ever since linguistic algorithm was successfully applied in the control of industrial process [5], the use of fuzzy logic control has proliferated. Such an approach is already the basis of many consumer products that make use of fuzzy rule-based control methodology.

It is well established that humans solve problems based on heuristics or rules of thumb. One characteristic feature of heuristic rules in the sense of humans' reasoning is that usually, the number of rules used to solve a particular problem is relatively small. Another feature of the rules tends to be generality. Each rule generalizes situations or circumstances into a form of linguistic association. For example, the statement "people who are obese, smoke, drink and do not exercise have high risk of suffering from heart attack". Here, we associate the class of people with high risk of heart attack to their physique and unhealthy lifestyle. Although not explicitly stated, such a rule may further imply that people who are slim, do not smoke and drink will have a low risk of heart attack. It seems that humans are able to work around with very few rules, without sacrificing their capability in problem-solving. However, this may be arguable for problem domains which require highly specialized knowledge. In our view, most fuzzy systems although successful in tackling the problems they are designed for, have not been able to capture the essence of "heuristics" in the sense of human oriented reasoning.

Existing applications or products on the implementation of silicon-based fuzzy reasoning usually rely on some form of microprocessor system based algorithms or special hardware architectures to replicate the exact fuzzy inference reasoning mechanisms. This may prove indispensable in control applications that require real-time operations where the domain rules are well-tuned. However, it is observed that most successful demonstration of fuzzy control systems rely on a large number of rules. In the true sense of fuzzy heuristics, we believe that this should not be so. To clarify this further and to ease further discussion, we define domain-specific fuzzy heuristics to be rules of thumb with the following salient characteristics:

- relatively few simple and abstract rules;
- with no precise but rather subjective definitions of concepts;
- not affected by nuances of humans' emotion.

In this paper, we describe architectures of fuzzy processors which are flexible to handle reasoning involving fuzzy heuristics that may require complex inferencing method. We will first give a brief theoretical overview before describing the simplified architecture of the fuzzy processor. Methods of incorporating parallelism to enhance the speed of inference will be briefly discussed. To demonstrate the approach comprehensively, we will make use of the practical example of a cart-pole balancing system based on the notion of fuzzy heuristics as mentioned earlier.

2. Theoretical Overview

Fuzzy control algorithms can be specified as *if-then* rules which may consists of one or more left-hand side antecedents and a right-hand side consequent. For example, a two-antecedent rule can be written as "*if (a and b) then c*". The implied fuzzy meaning of rules are in the premises *a*, *b* and *c* which can be implicitly represented as fuzzy sets *A*, *B* can *C* in some universe *U*, *V* and *W* respectively. If *a'* and *b'* denote two known premises (or facts) similar to *a* and *b* respectively, the rule of inference for computing *c'* can be written as follows [8]:

$$C' = A' \circ (A \to C) \wedge B' \circ (B \to C) \tag{1}$$

where C' denotes the deduced fuzzy set and A' or B' are fuzzy sets denoting some known facts. In the above expression, "o" is used to denote max-min composition, "$\wedge$" denotes min operation and "$\to$" denotes the bivariate relation also known as fuzzy implication. Except for the "$\to$" operation, we feel that there is general agreement towards the other operators. Many fuzzy processors, for example [7], adopt the following relation to compute the conditional proposition or if-then rule:

$$R_{A \to C} = A \times C \tag{2}$$

where A and C represent two fuzzy sets and "$\times$" denotes Cartesian product. $R_{A \to C}$ can be referred to as a fuzzy implicational relation. To compute the conclusion based on such a relation, the max-min composition described by equation (1) can be written as follows:

$$\begin{aligned} \mu_{C'}(w_k) = \; & [\vee_{\forall u \in U}(\mu_{A'}(u_i) \wedge \mu_A(u_i)) \wedge \mu_C(w_k)] \wedge \\ & [\vee_{\forall v \in V}(\mu_{B'}(v_j) \wedge \mu_B(v_j)) \wedge \mu_C(w_k)] \end{aligned} \tag{3}$$

The symbol $\mu_A(u_i)$ denotes the membership value of element u_i in the universe U as defined by the fuzzy set A. This method of inference is not difficult to implement in view of the fact that the computations involved can be realized by simple min and max functional blocks which is easily achieved in hardware.

For a fuzzy system of n rules, we can represent it symbolically as

$$c' = @\ (c_1', c_2', \ldots, c_n') \tag{4}$$

where c_i' denotes the conclusion derived by firing the i^{th} rule computed from equation (1) and @ denotes the aggregation operator. The mode of aggregation dictates how the consequents from firing all the n rules are combined to obtain the final conclusion. Here again, to our knowledge as reflected by the frequent usage of it, there appears to be general tendency towards a disjunctive mode of aggregation which can be easily realized by means of a max function block. An obvious alternative would be a conjunctive mode of aggregation realizable by a min function block. One should realize that in fuzzy reasoning, many aggregating operations are possible.

3. Architecture of Fuzzy Processor

In section 2, we mentioned that the relation $R_{A \to C}$ as in equation (2) can be realized by means of a min function block. Because of its simplicity, it is practical for on-chip computation of the fuzzy implication. However, if we were to adopt the following relation as described in Mizumoto [6]:

$$R_{A \to C} = (A \times W <g> U \times C) \wedge (\neg A \times W <g> U \times \neg C) \qquad (5)$$

where <g> is defined as:

$$\alpha <g> \beta = \begin{cases} 1 & \alpha \leq \beta \\ b & \alpha > \beta \end{cases} \qquad (6)$$

It is clear that on-chip computation of the fuzzy implication may become significantly more complicated. An alternative would be to pre-compute the relation and store it as matrices in the processor's memory [4]. With the pre-computed matrices, the general block architecture of the processor is as shown in Figure 1. Both the min and max blocks compute the max-min composition for every rule. The lower max/min block realizes the aggregation of conclusions derived from firing all the rules. We distinguish a fuzzy register as register with storage for a fuzzy set. A normal register stores 4 bits membership value.

It is clear that the hardware configuration is not affected by the mode of implication used. Another consideration is the universe size of the fuzzy concepts. The capability of fuzzy reasoning is not sacrificed by compelling designers to define fuzzy concepts of a uniform universe size. From a hardware point of view, this can be easily achieved by loadable counters which can be presetted prior to inferencing.

Alternatively, one can achieve the flexibility by dynamically setting the counters as and when the universe size changes. This can be done by incorporating a header declaration with information about the number of rules, the universe size of the consequent, the number of antecedents followed by the universe size of each of the antecedents. Subsequently, before each matrix is accessed, an antecedent identifier followed by a flag to declare the logical mode to be used in combining with the next matrix. This can be best illustrated by means of the example in the next section where the so called relation matrices are shown in Table 1.

So far, our architectural descriptions have not included fuzzification of input and defuzzification of output. It is assumed that inputs are fuzzified and loaded onto a fuzzy input register to be used for inferencing. The process of defuzzification is concurrent with the inferencing process. As and when the membership value of the consequent is derived by the inferencing unit, the intermediate terms that are necessary for defuzzification are computed and stored in a fuzzy register. When the final inferencing step is completed, the intermediate terms are used to obtain the final defuzzified outcome.

4. Cart-Pole Balancing Problem

In this section, we describe the practical problem of a control system that make use of our notion of control heuristics. This problem is commonly referred to as the inverted pendulum problem. To closely reflect on practical reality, we refer to it as the cart-pole system. The cart is rigidly hinged such that it can fall either to the left for a negative θ or right for a positive θ. To counter a positive angle of the pole, a movement towards the right is required to bring the pole to a vertical position. Likewise, a movement to the left is needed to compensate for a negative angle. In addition to θ, the angular velocity θ' is also used to determine the direction of movement of the cart.

The inverted pendulum problem is a common example to illustrate control algorithms including fuzzy control. Many studies using fuzzy controllers (for example [1], [2] and [3]) make use of 9 or more rules. To our knowledge, no formal method exists for determining the number of rules or fuzzy linguistic values. For example, how do we know that the linguistic variable angle should take on the linguistic values positive_big, positive_small, zero, negative_small and negative_big. This being so, designers of fuzzy controllers usually resort to partitioning of the universe to manageable range to be assigned with a fuzzy term. This is probably done intuitively or based on precedence set by other successful fuzzy control algorithms. It is not difficult to notice that as the number of linguistic values increases, the number of rules tend to increase significantly making it tedious to achieve a functional controller. This is (although arguable) tantamount to exhaustive declaration of all possible associations of linguistic values of the antecedents and that of the consequent.

To illustrate the notion of heuristics advocated in this paper, we imagine intuitively the rules borne in the human mind to solve this problem. Although experts may use different set of rules to solve similar problem just as effectively, it is likely that in this case, 2 rules may be sufficient to generalize the cart-pole balancing system. Consider the following rules:

if *(angle and angular velocity are positive)*
then *(move cart to the right)*
else_and
if *(angle and angular velocity are negative)*
then *(move cart to the left)*

The above rules generalizes the situation when force needs to be applied to the cart to stop the pole from falling. The degree of movement of the cart can be realized by supplying the proper torque to the motor.

Figure 2 shows the graphical representation of the fuzzy terms negative and positive for angle (θ), angular velocity (θ') and force (F). With these definitions of the fuzzy terms, 4 fuzzy relation matrices can be generated based on the relation described by equations (5) and (6) with the necessary declarations defined earlier in the previous section. Two of the matrices are shown in Table 1 for illustration.

5. Parallelism

To enhance the computation speed, it is possible to incorporate parallelism in the processor. If we conservatively estimate that each min or max block requires 200 gates and each register is realized using 40 gates (assuming that membership is coded as 4 bits value and each bit of the register needs 10 gates), the inference unit shown in Figure 1 can be realized with less than 1000 gates. Hence for a VLSI chip of 50,000 gates capacity, a reasonable estimate will suggest that we can accommodate more than 48 inference units to allow for simultaneous processing of 48 relation matrices. We refer to this as rule parallelism since all the rules are fired in a parallel manner. This is depicted in Figure 3 showing N-rules parallelism. The symbols R_1, R_2, ..., R_N represent pre computed relation matrices.

Another alternative is to incorporate parallelism during the firing of each rule. For example, if the matrix is $M \times N$ where M is the antecedent's universe size and N corresponds to the consequent's universe size, we can make use of M min blocks followed by $(N-1)$ max blocks to achieve the max-min composition. We refer to this as max-min parallelism. Figure 4 shows the general block architecture to illustrate max-min parallelism. The symbols u_{i1}, ..., u_{im} denote membership values of a column of the matrix. Complete processing of a matrix occurs when i equals N.

6. Conclusions

In this paper, we have discussed the issue of hardware architecture for general purpose fuzzy processors which is independent of the inference method used. Although there are many successful fuzzy controllers to show that simple inference method of the Mamdani [5] type is sufficient to tackle a myriad of control problems, we contend that real-life problems may demand more complicated inference method. This may result in far less number of rules than it ordinarily would have. We made used of an inverted pendulum control problem to illustrate this point. The motivation for this paper is to describe the architecture of what we perceive as general purpose fuzzy processors realizable on a VLSI chip. If memory is incorporated in the fuzzy processor, the simple yet powerful architecture allows for single chip fuzzy reasoning system somewhat similar and analogous to a micro controller. Although the architectures described demand a greater number of max or min computations, it is feasible to enhance the inference speed by re-configuring the processor to incorporate rule parallelism or max-min parallelism.

Acknowledgment

The authors wish to acknowledge the hard work of their undergraduate students Chen Yeang Tat and Cheng Kong San in fabricating the cart-pole balancing system as their final year project.

References

[1] Kosko B., "Neural Networks and Fuzzy Systems - A Dynamical Systems Approach to Machine Intelligence", Prentice-Hall International Incorporated, Englewood Cliffs, New Jersey. 1992.

[2] Lee C.C., "A Self-Learning Rule-Based Controller Employing Approximate Reasoning and Neural Net Concepts", International Journal of Intelligent Systems, Vol. 6, pp. 71-93, 1991.

[3] Leong J.Y., Lim M.H. and Lau K.T., "Development Strategy for Fuzzy Controllers", International Conference on Computer Automation, Robotics & Vision, Singapore, Sept. 1991.

[4] Lim M.H. and Takefuji Y., "Implementing Fuzzy Rule-Based System on Silicon Chips,"IEEE Expert, February 1990.

[5] Mamdani E.H. and Assilian S., "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller," (Mamdani, E.H. and Gaines, B.R. eds.), London: Academic Press, pp. 133-148, 1981.

[6] Mizumoto M. and Zimmerman H., "Comparisons of Fuzzy Reasoning Methods," Fuzzy Sets and Systems, 1980.

[7] Togai M. and Watanabe H., "A VLSI Implementation of Fuzzy Inference Engine: Towards an Expert System on a chip", Information Sciences, Vol. 38, pp. 147-163, 1986.

[8] Zadeh L.A., "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," IEEE Transactions on Systems, Man and Cybernetics, 3:28-44, 1973.

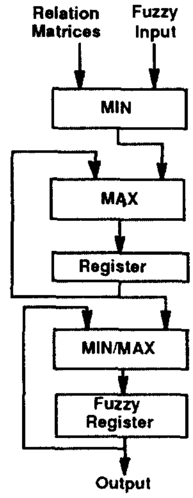

Figure 1: Block architecture of an inference unit.

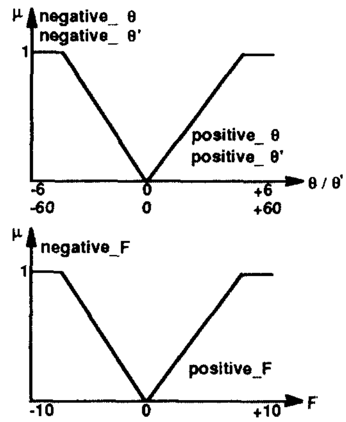

Figure 2 : Memberships of fuzzy terms.

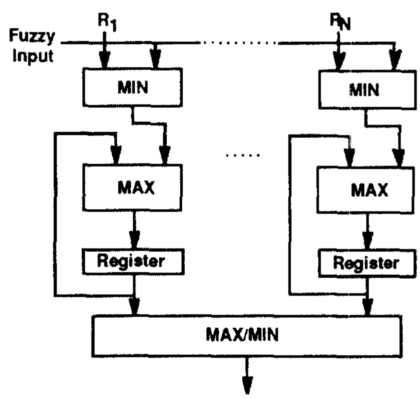

Figure 3: Rule parallelism.

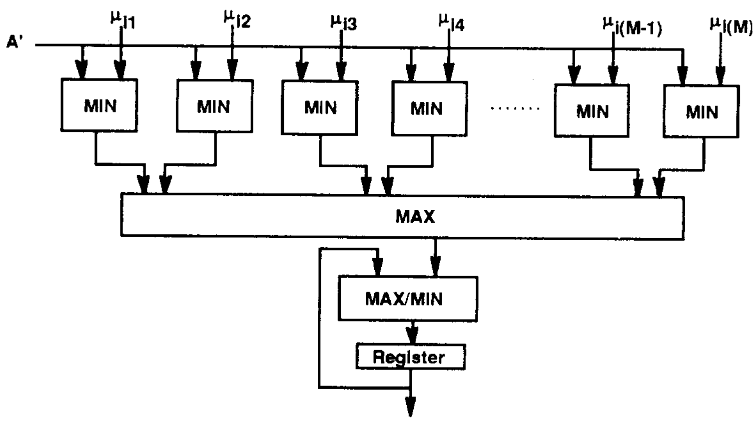

Figure 4: Max-min parallelism.

~2 11 2 13 13

~1 &

1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.6	0.4	0.2	0.0
1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.6	0.4	0.2	0.0
1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.6	0.4	0.2	0.0
1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.6	0.4	0.2	0.0
1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.6	0.4	0.2	0.0
1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.6	0.4	0.2	0.0
1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.6	0.4	0.2	0.0
0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.6	0.4	0.2	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.6	0.4	0.2	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.4	0.4	0.2	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.4	0.6	0.2	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.4	0.6	1.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.4	0.6	0.8	1.0

~2 A

1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.6	0.4	0.2	0.0
1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.6	0.4	0.2	0.0
1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.6	0.4	0.2	0.0
1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.6	0.4	0.2	0.0
1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.6	0.4	0.2	0.0
1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.6	0.4	0.2	0.0
1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.6	0.4	0.2	0.0
0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.6	0.4	0.2	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.6	0.4	0.2	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.4	0.4	0.2	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.4	0.6	0.2	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.4	0.6	1.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.4	0.6	0.8	1.0

Table 1: Implicational relation matrices.