LETTER

Center-of-Gravity Defuzzification without Multiplication

Stamatis BOURAS† *and* **Yannis TSIVIDIS**††, *Nonmembers*

SUMMARY An alternative defuzzification technique for center of gravity calculation is proposed. The technique allows evaluation of fuzzy inferences without the use of multiplication. In fuzzy logic controllers with many outputs, the proposed scheme reduces the circuit complexity compared with other implementations.

key words: *fuzzy logic, defuzzifiers*

1. Introduction

One of the main bulding blocks in fuzzy controllers is the defuzzifier. The most common defuzzification method is based on a "center of gravity" calculation [1], which produces a "crisp" output value, y_{crisp}, based on the following calculation:

$$y_{crisp}=\frac{\int_0^a w\cdot\mu(w)\,dw}{\int_0^a \mu(w)\,dw} \tag{1}$$

where $\mu(w)$ is the fuzzy inference function, w is its argument, and a is the maximum value of w. In analog fuzzy controllers, the continuous function calculations in (1) can be implemented directly. Digital controllers use discretized versions of (1) [2], and these are often used also in analog controllers [3]–[7]. The calculation is then done according to:

$$y_{crisp}=\frac{\sum_{i=0}^{n_a} w_i\cdot\mu(w_i)}{\sum_{i=0}^{n_a}\mu(w_i)} \tag{2}$$

where w_i are the values of n_a+1 discrete samples of w. In this letter, instead of Eqs. (1) and (2) we propose an alternative mathematical computation of the center of gravity that uses no multiplication and can, in certain cases, result in reduced hardware complexity (for parallel calculation) or reduced computation time (for serial calculation).

Manuscript received October 15, 1996.

† The author is with the Division of Computer Science, National Technical University of Athens, Heroon Polytechniou 9, Zografou 15773 Athens, Greece.

†† The author is with the Department of Electrical Engineering, Columbia University, New York, NY 10027, USA.

2. Principle

The complexity of defuzzifier realizations based on (1) becomes apparent if one considers an application where a fuzzy logic controller with many outputs is needed, in order to drive many actuators. In that case, the calculation is done as shown in general form in Fig. 1. An analog controller based on this diagram is given in [8], [9]; in that implementation, w is swept versus time and all calculations are done as a function of time, with the final value being produced at the instant T at which $w(T)=a$. As the product $w\cdot\mu_i(w)$ is unique to each distinct function μ_i, as many multipliers as there are outputs are needed, if all defuzzifications are to be done in parallel. This results in increased power dissipation and chip area. If a single multiplier is time-shared, increased calculation time results. To circumvent these problems, we manipulate (1) in order to eliminate the product $w\cdot\mu(w)$ in it. If we use a change of variables $y=w^2/a$ in the numerator, Eq. (1) becomes:

$$y_{crisp}=\frac{\frac{a}{2}\int_0^a \mu(\sqrt{a\cdot y})\,dy}{\int_0^a \mu(y)\,dy} \tag{3}$$

This equation gives the same output as (1). The corresponding discretized version of this equation is:

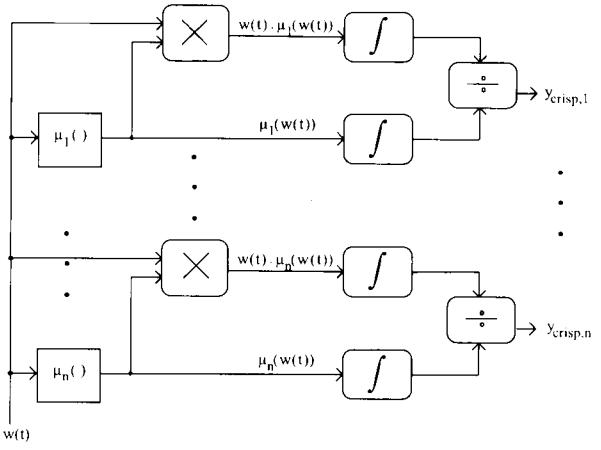

Fig. 1 Conventional defuzzification, for a controller with many outputs.

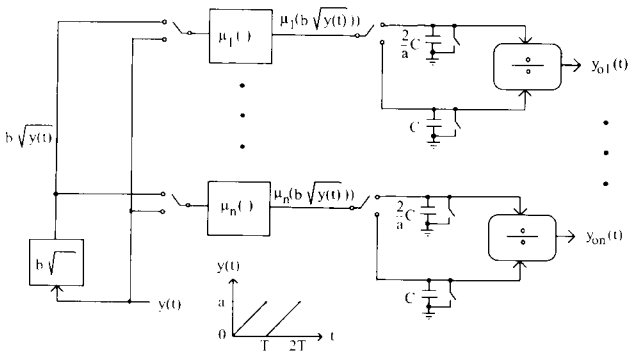

Fig. 2 Analog implementation example using the proposed technique, for a controller with many outputs.

$$y_{crisp} = \frac{\frac{a}{2} \cdot \sum_{i=0}^{n_a} \mu(\sqrt{a} \cdot y_i)}{\sum_{i=0}^{n_a} \mu(y_i)} \tag{4}$$

In these two equations, the multiplication has been replaced by a square root operation. The argument of the inference function in the numerator is different from that in the denominator, and thus must be generated separately. Note, however, that if many defuzzifiers are needed this argument is common to all of them, and thus the calculation of the square root can be shared.

3. Hardware Example

An analog implementation of Eq. (3) is shown in Fig. 2. The input is a ramp $y(t)$. The square root function is generated only once. The square root and divider circuits can be implemented using only a few MOS transistors each, as shown in [10]. The inference function generators have a current output [4], [6]-[9], which is directly integrated by feeding it to capacitors. The capacitances are properly scaled in order to take into account the different multiplying constants in front of the integrals in (3). The switches in parallel with the capacitors are used to reset their voltages to zero before a new calculation starts. One inference function generator is used for each defuzzifier output, and is shared for the calculations of the numerator and the denominator in (3). The switches connected to the inference function generators move in the same direction. The center of gravity calculation is done within two sweep periods. In the first sweep period ($[0, T]$) the numerator of Eq. (3) is calculated for each path. At the end of this period, the switches move downwards and the value of the numerator is stored on the top capacitor of each path. In the second sweep period ($[T, 2T]$) $y(t)$ is swept again, and the denominator is calculated. The desired output y_{crisp} is obtained at the end of the second sweep period and is $y_{crisp} = y_0(2T)$. Thus, the penalty in calculation time in comparison with the system of Fig. 1 is a factor of 2, whereas the complexity is significantly lower for $n \gg 2$. If, in order to achieve comparable complexity in the system of Fig. 1, it is attempted to use a single, time-shared multiplier, then the calculation time becomes larger than that for Fig. 2 by the factor $n/2$, where n is the number of defuzzifier outputs.

4. Conclusion

A new approach to the computation of the center of gravity in defuzzifiers has been proposed, in which no multiplication is required. This can result in reduced complexity and/or calculation time in controllers with multiple defuzzifier outputs. An analog implementation, presented as an example, shows the efficiency of the technique proposed.

References

[1] L. A. Zadeh, "Fuzzy Sets," Information and Control, vol. 8, pp. 338-353, New York: Academic Press, 1965.

[2] H. Watanabe, W. Dettloff, and K. Yount, "A VLSI fuzzy logic controller with reconfigurable, Cascadable Architecture," IEEE J. Solid-State Circuits, vol. 25, pp. 376-382, April 1990.

[3] T. Yamakawa, "A fuzzy inference engine in nonlinear analog mode and its application to a fuzzy logic control," IEEE Trans. Neural Networks, vol. 4, no. 3, pp. 496-522, May 1993.

[4] T. Kettner, K. Shumacher, and K. Goser, "Realization of a monolithic analog fuzzy logic controller," IEEE Proc. ESSCIRC 1993 Sevilla, pp. 66-69.

[5] J. Fattaruso, S. S. Mahant-Shetti, and J. B. Barton, "A fuzzy logic inference processor," IEEE J. Solid-State Circuits, vol. 29, no. 4, pp. 397-402, April 1994.

[6] A. Rodriguez-Vazquez and F. Vidal, "Modular design of adaptive analog CMOS fuzzy controller chips, "Proc. ESSCIRC, pp. 122-125, Sept. 1995.

[7] J. L. Huertas, S. Sanchez-Solano, I. Baturone, and A. Barriga, "Integrated circuit implementation of fuzzy controllers," Proc. ESSCIRC, pp. 130-133, Sept. 1995.

[8] H. Tang and H. C. Lin, "Defuzzifier circuits using resonant tunneling diodes," Proc. IEEE Symp. Circuits Syst., pp. 981-984, May 1995.

[9] S. Bouras, M. Kotronakis, K. Suyama, and Y. Tsividis, "Mixed analog-digital fuzzy logic controller with continuous fuzzy inferences and defuzzification," in IEEE Transactions on Fuzzy Systems.

[10] R. J. Wiegerink, "Analysis and synthesis of MOS translinear circuits," Kluwer Academic Publishers, Boston, 1993.