

Performance Aspects of Fuzzy Logic Controlers with Improved Architecture

Doru Todinca¹, Radu-Emil Precup²

¹ Department of Computers, University Politehnica Timisoara, e-mail: doru.todinca@ac.upt.ro

² Department of Automation, University Politehnica Timisoara, e-mail: radu.precup@ac.upt.ro

*Abstract: Fuzzy Logic Controlers (FLCs) are circuits that implement fuzzy inference. Based on the manner in which the fuzzy inference is performed, digital FLCs can be classified as **classic** and **improved** [patyra]. In this paper we investigate the performance of the improved FLC during the control process.*

Keywords: Fuzzy logic controller, improved architecture

1 Introduction

Togai and Watanabe [11] realized the first hardware implementation of a fuzzy logic controller (FLC). Since then, many FLCs have been implemented, using different analog and digital technologies and different architectures. In this paper, we focus on the performance of digital FLCs. Because of the large variety of FLCs' implementations, it is difficult to compare their performance.

Patyra [5] presented a systematic approach to performance analysis of digital fuzzy logic circuits, focusing on their hardware costs and time performance (input-output delay). Based on [5], the main characteristics of FLCs are:

- The number of input variables (K), which is also the number of FLC's inputs.
- Number of outputs (L). In our investigations we consider $L=1$, which does not affect the generality of the results.
- Maximum number of terms (membership functions) for each of the input linguistic variables (MB_{IN}).
- Maximum number of terms for the output linguistic variables (MB_{OUT}).

- Maximum number of rules (N). It results that $N=(MB_{IN})^K$.
- Number of binary vectors used to represent the domain of the linguistic variables (i.e., of their membership functions) (n).
- Number of bits in a single binary vector (m).
- Input-to-output delay time (Θ_{IN-OUT}).

Two architectures are analysed for FLCs: the “classic” architecture and the “improved” architecture, which has better time performance (Θ_{IN-OUT}), because a part of the fuzzy inference is processed off-line, i.e. before the input values are presented at the FLC’s inputs.

However, the approach from [5] takes into account only the performance of the FLCs as digital circuits, without considering their efficiency in the control process. In [9], [2], [1] we extended the analysis from [5], by investigating the performance of different FLC types during the control process. Also, we determined which of the parameters that characterise a FLC have an important influence on its performance in a control application.

The study was performed through digital simulation, using the hardware description language VHDL. The architectures investigated were the classic FLC [5], improved FLC [5], [4], and the simplified FLC [3], [10]. The parameters with a decisive influence on the performance of the control operation were the number of binary vectors used to represent the fuzzy linguistic variables n , and Δ – the degree of fuzzification for inputs. If $\Delta=0$, the input value (the fact) is not fuzzified, otherwise, if $\Delta>0$, the value of the input is transformed into a triangular fuzzy set, the base of the triangle being equal to 2Δ .

From the three architectures that were investigated in [9], [2], [1], the performance of the improved architecture in the control process were extremely poor, except for the case when the FLC had only one input ($K=1$), when the improved FLC and classic FLC performed identically.

In this paper we investigate the performance of the improved FLC. This investigation is important because, on the one side, this architecture has much better time performance, as a digital circuit (the Θ_{IN-OUT} is much smaller than for classic FLC), being also possible to apply parallel algorithms in order to improve its performance [4], but on the other side, as shown in [9], [2], [1], its performance in the control process are very poor.

2 Fuzzy Inference

2.1 Classic Method

A fuzzy inference is a relation $B=A \circ R$, where R is the fuzzy implication $A \rightarrow B$. The premises A can be composed, being expressed in the form $A=A^1 \text{ AND } A^2 \text{ AND } \dots \text{ AND } A^K$, in the case when there are K input variables.

According to [5], for a FLC with K inputs (K input variables) and a maximum number of N fuzzy rules, if A_i^j , are subsets of A^j , $i=1, \dots, N$ and $j=1, \dots, K$, then the set of N fuzzy rules is:

$$\begin{aligned}
 R_1: & \text{ IF } A_1^1 \text{ AND IF } A_1^2 \text{ AND } \dots \text{ AND IF } A_1^K \text{ THEN } B_1, \text{ ALSO} \\
 R_2: & \text{ IF } A_2^1 \text{ AND IF } A_2^2 \text{ AND } \dots \text{ AND IF } A_2^K \text{ THEN } B_2, \text{ ALSO} \\
 & \dots \dots \dots \\
 R_i: & \text{ IF } A_i^1 \text{ AND IF } A_i^2 \text{ AND } \dots \text{ AND IF } A_i^K \text{ THEN } B_i, \text{ ALSO} \quad (1) \\
 & \dots \dots \dots \\
 R_N: & \text{ IF } A_N^1 \text{ AND IF } A_N^2 \text{ AND } \dots \text{ AND IF } A_N^K \text{ THEN } B_N.
 \end{aligned}$$

In this case the fuzzy relation is

$$B = (A^1, A^2, \dots, A^K) \circ R \quad (2)$$

The overall fuzzy relation R is

$$R = \bigcup_{i=1}^N R_i \quad (3)$$

The input observations (facts) being A_1, A_2, \dots, A_K , the result $B_1 = (A_1, A_2, \dots, A_K) \circ R$ of the overall relation is obtained by applying the compositional rule of inference:

$$B_1 = (A_1, A_2, \dots, A_K) \circ \bigcup_{i=1}^N R_i = \bigcup_{i=1}^N (A_1, A_2, \dots, A_K) \circ R_i \quad (4)$$

Considering the rule R_i , the action (the consequence of the rule) is given by

$$B_{1i} = (A^1, A^2, \dots, A^K) \circ R_i \quad (5)$$

Using the MAX-MIN operation for the composition rule, after transformations, the membership function of B_{1i} is given by the relation:

$$\mu_{B_{1i}}(b) = \text{MIN}(\Omega_i, \mu_{B_i}(b)); \text{ where } b \in B_1 \quad (6)$$

with

$$\Omega_i = \text{MIN}(\text{MAXMIN}(\mu_{A1}(a_1), \mu_{A1}(a_1)), \dots, \text{MAXMIN}(\mu_{AK}(a_K), \mu_{A1^k}(a_K))); \quad (7)$$

with $a_1 \in A1, \dots, a_K \in AK$

The final action, B1, is the maximum between B1₁, B1₂, ..., B1_N:

$$B1 = \bigcup_{i=1}^N B1_i = \text{MAX}(B1_1, B1_2, \dots, B1_N) \quad (8)$$

2.2 Improved Method

This method is based on rule decomposition into sub-relations, which means that we can assume that each rule R_i can be de-composed into K sub-relations, one for each input:

$$R_i^k = A_i^k \times B_i, \quad (9)$$

where i=1,2,..., N denotes the rule number, and k=1,2,..., K, denotes the input variable.

The MIN operator is used to compute the Cartesian product of two variables, one variable from premises, and the other from conclusion, which means that:

$$R_i^k(a_k, b) = \text{MIN}(A_i^k(a_k), B_i(b)) \quad (10)$$

The overall k^{th} sub-rule for the combination of the kth input A^k and the output variable B is obtained by the union of all the N sub-relations:

$$R^k = \bigcup_{i=1}^N R_i^k \quad (11)$$

$$\text{or } R^k = \text{MAX}(R_1^k(a_k, b), R_2^k(a_k, b), \dots, R_N^k(a_k, b)) \quad (12)$$

For a set of inputs (A1, A2, ..., AK), the output B1 can be computed using the MIN-superposition of all K relations between the input Ak and the rule R^k:

$$B1 = \text{MIN}[A1 \circ R^1, A2 \circ R^2, \dots, AK \circ R^K] \quad (13)$$

or

$$B1 = \text{MIN}\{\text{MAXMIN}[A1, R^1(a_1, b)], \dots, \text{MAXMIN}[AK, R^K(a_K, b)]\} \quad (14)$$

3 Fuzzy Logic Controllers

3.1 Classic Architecture

The classic architecture of the FLC follows the set of equations (1) – (8) and is presented in [5]. We will use it in this paper only as a reference for the performance of the improved FLC in control applications.

3.2 Improved Architecture

Figure 1 presents the architecture of the improved FLC, the DISO (Double Input Single Output) case. For the more general MISO (Multiple Input Single Output) case the structure is similar.

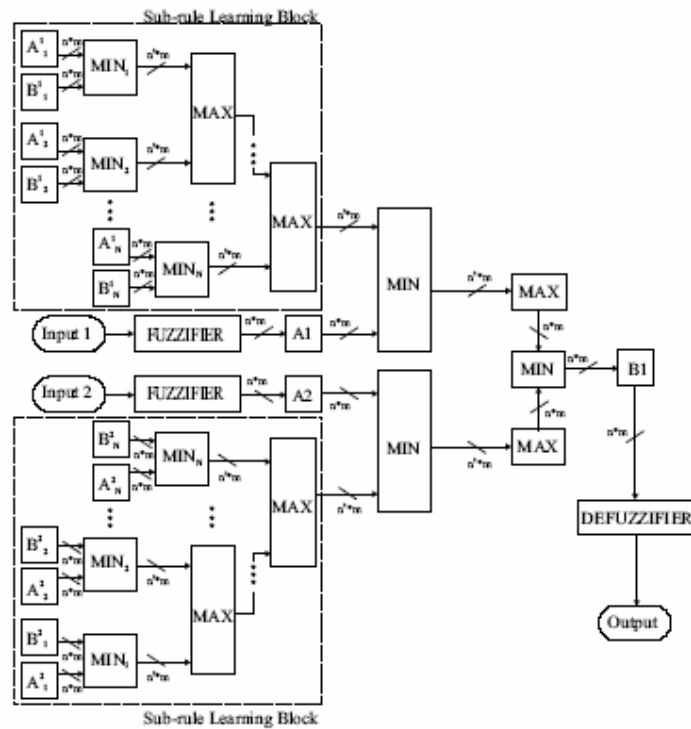


Figure 1
FLC improved

For each input, the sub-rule learning block consists of N MIN circuits that compute the Cartesian product between each A_i^k and each B_i , according to relation (10). The output of each circuit is a $n*n$ vector of m bits that feeds the tree of $(N-1)$ MAX circuits, obtaining the partial rule R^k , according to (12). This part of the fuzzy inference can be processed off-line, i.e., before the value of the observation (fact) appears at the inputs of the FLC.

Each input can be fuzzified (according to the value of the parameter Δ , obtaining the fuzzy set A^k). The MAXMIN operation between A^k and the sub-rule R^k is realised by the MIN circuit that has an input of $n*n*m$ bit vectors and another input on $n*m$ bits, and the output is on $n*n*m$ bits, and by the MAX circuit that obtains an $n*m$ bits conclusion from the n^2*m bit vector represented by the output of the MIN circuit.

The value of B1 from (14) is obtained by a final MIN operation between the partial rules R^1, \dots, R^K (in Figure 1, $K=2$). Then B1 is defuzzified and the resulting crisp value appears at the output of the FLC.

4 Performance of the Improved FLC in Control Applications

4.1 Simulation Setup

4.1.1 FLC Description

The classic and the improved architectures of digital FLCs described in [5] have been modeled in the hardware description language VHDL. VHDL was used because it can model both the hardware aspects of the FLCs, and the behavioral, more abstract aspects of the controlled processes. It is important to mention that the FLCs' architectures from [5] do not refer to particular implementations of FLCs, but they are very general. By using VHDL packages containing the functions that describe the FLC architectures, we have obtained very flexible and general descriptions of FLCs. Some other packages are used for the linguistic variables and fuzzy rules, while the controlled process is described by a VHDL entity called *black_box*, that is completely independent from the FLC's description. Detailed description of the VHDL implementations of the FLCs can be found in [1], [6].

4.1.2 Performance Indicators

In [5], only the performance of the FLCs as digital circuits are taken into account, namely the hardware cost of the circuits and the input-output delay Θ_{IN-OUT} . From those point of view, the improved FLC clearly outperforms the classic architecture.

In order to have a more conclusive view on the FLCs' performance, in [9] we have tested the FLCs during control applications, using non-linear processes that are difficult to control, being necessary a large number of control steps in order to control them. First process is described in [8], and is a process with one input and one output. The classic and the improved FLCs provide identical results, which is a test of correctness of the VHDL descriptions.

The most interesting results were obtained for a non-minimum phase system described in [7]. The process has one reference input w and a disturbance input v . The control error is $e=w-y$ is, while y is the controlled output and u is the control signal. The FLC uses the control error e and its first and second increment as input variables, while the output of the FLC is the change of the controlled output u . Figure 2 [9] shows the system response (i.e. the evolution of y versus time t expressed in number of control steps) with respect to the unit step variation of reference input w .

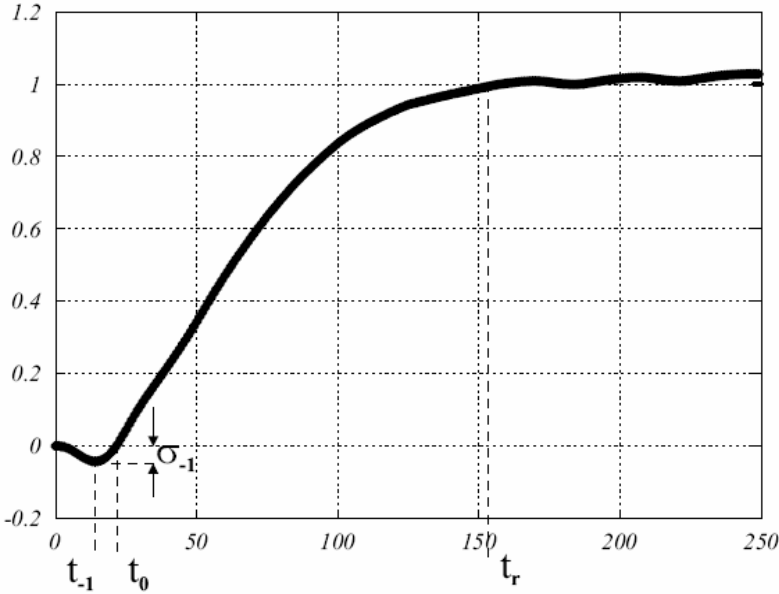


Figure 2
System response y with respect to the unit step modification of the reference input

The parameters of interest for the performance of the FLC are:

- the downshot (σ_{-1});
- the moment of σ_{-1} (t_{-1});
- the return time (t_0);
- the settling time (t_r).
- The deviation $dev = |y - w|$, considered when the system enters the steady state regime.

The deviation expresses the control quality and in the ideal case it is zero. We consider the control quality acceptable if $dev < 2\%$.

4.2 Results

4.2.1 Unmodified Improved FLC

The simulation results presented in this work (see also [6]) were obtained for the following values of the parameters: $n=64$, $m=4$, $A=0$, $K=3$, $L=1$ (three inputs and one output). The reference input has a unit step variation at the moment $t=0$, and the system output y is plotted in figure 3 against the control time, expressed in number of control steps, for 1000 control steps. It is clear from the figure that the control performance of the improved FLC are very poor, the values of the controlled output y being much higher than 1 (y is between 1.4 and 1.6, with high amplitude oscillations).

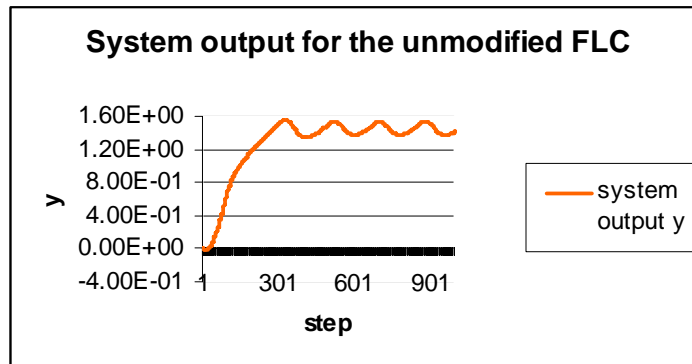


Figure 3

The results of the improved FLC during the control process

The explanation for these very bad performance lies in the fact that the initial fuzzy rules were written in the form “if premise1 and premise2 and premise3 then conclusion”, which are suited for the classic inference, and then projected on the 3 input variables, [5], obtaining: three rules: “if premise1 then conclusion”, “if premise2 then conclusion” and “if premise3 then conclusion”, according to relation (9) from section 2.2. Only the projected rules can be implemented in the improved FLC. Because of this transformation, an input fact activates too many rules, such that the effect of the rules compensate each others and the output of the FLC is very often close to zero.

4.2.2 Modified Improved FLC

In order to solve this problem, we decided to modify the improved FLC, by adding a functional block that determines, for a set of input facts, which input terms are activated, and then, which terms in conclusions are activated in the original fuzzy rules (as used by the classic FLC [9], [1]). Then, a MINIMUM operation between the actual conclusion of the FLC and the activated terms is performed before defuzzification. More details can be found in [6]. The output y of the system for the modified improved FLC is shown in figure 4:

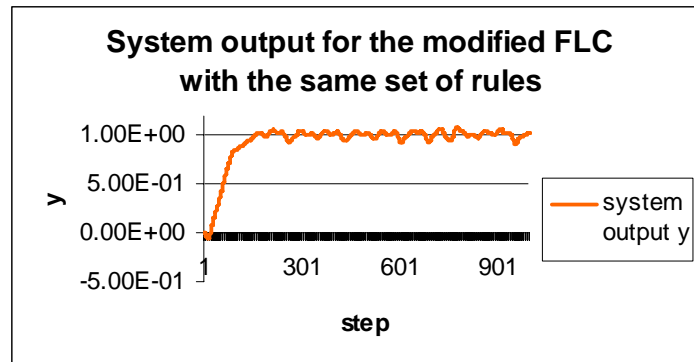


Figure 4
System output y for the modified FLC

The results are much better now, but the values of parameter dev are too high, which means that the output of the FLC in the steady state region oscillates too much.

4.2.3 Modified Improved FLC when the Fuzzy Rules Change

In order to reduce the oscillations in the steady state regions, we decided to change the set of rules used by the FLC. From the point of view of a digital circuit, this corresponds to a change of the content of the memory during the functioning of

the circuit. The condition for changing the rules is that $dev < 2\%$ and the number of control steps to be higher than a certain value. Figure 5 presents the output of the system in the case of the modified FLC when the change of the rules take place after the step 200.

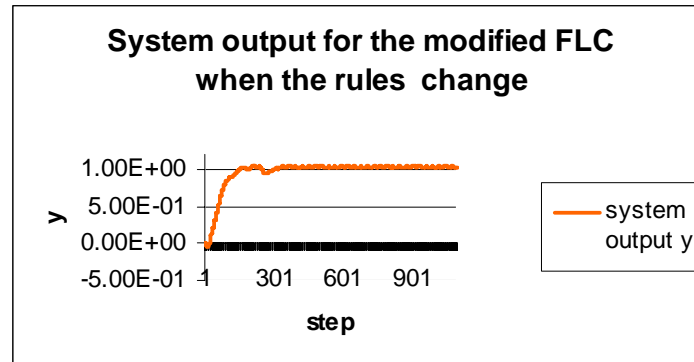


Figure 5

The output of the system when the fuzzy rules are changed after 200 control steps

Now the improved FLC performs better, the value of the deviation dev being within the 2% limit. From Figure 5 it can be observed that the output of the system, y , shows some oscillations before entering the steady state region.

Conclusions

This paper continues our previous study from [9], about the performance of different fuzzy logic controllers. Here we have investigated the performance of the improved FLC digital architecture in control applications. Despite the fact that the improved FLC shows better time performance than the classic FLC, as a digital circuit, its control performance are quite poor. The explanation is based on the fact that the transformation of the fuzzy rules in order to be suitable for the improved FLC reduces the accuracy of the control process. This aspect is more visible if the premises of the rules are composed and consists of a number of subpremises higher than 2 (in our case there are 3 subpremises).

We used some ad-hoc techniques in order to overcome these problems, techniques that include the determination of the terms from premises that are activated by a certain input fact, and the change of fuzzy rules when the controlled process approaches the steady-state situation.

Further investigations are necessary in order to obtain more results about the performance of the improved FLC in control applications, investigations that include the use of the improved FLC in different control applications.

Acknowledgements

The authors express their gratitude to Ana Maria Bădulescu and Ruxandra Popet who implemented the VHDL code describing the classic and improved FLCs.

References

- [1] Ana Maria Badulescu: Studiul performantelor circuitelor fuzzy utilizand limbajul VHDL (Performance Study of Fuzzy Circuits using VHDL - in Romanian), Final year project, supervisor: Doru Todinca, Timisoara, June 1999
- [2] Ana Maria Badulescu, Doru Todinca, Radu Emil Precup: Performance Considerations of Fuzzy Logic Controllers Using VHDL, in Periodica Politehnica, Transactions on Automatic Control and Computer Science, vol. 44 (58), Timisoara, 1999, pp. 179-186
- [3] T. Chiueh: Optimization of Fuzzy Logic Inference Architecture in IEEE Computer, May 1992, pp. 67-71
- [4] J. L. Grantner: Parallel Algorithm for Fuzzy Logic Controller, in Fuzzy Logic: Implementations and Applications, edited by Marek J. Patyra and D. M. Mlynek, John Wiley & Sons Ltd. and B. G. Teubner, 1996, pp. 177-195
- [5] Marek J. Patyra: Design Considerations of Fuzzy Logic Controllers, in Fuzzy Logic: Implementations and Applications, edited by Marek J. Patyra and D. M. Mlynek, John Wiley & Sons Ltd. and B. G. Teubner, 1996, pp. 143-175
- [6] Ruxandra Popoț: Studiul si implementarea VHDL a unor circuite fuzzy neconventionale (Study and VHDL Implementation of Fuzzy Unconventional Circuits – in Romanian), Final year project, supervisor: Doru Todinca, Timisoara, June 2000
- [7] Radu Emil Precup: On the Parameter Adaptation of a Fuzzy Controller Meant for the Speed Control of Hydrogenerators, in Proceedings of Third EUFIT'95 European Congress, Aachen, Germany, 1995, Vol. 2, pp. 1105-1109
- [8] Stefan Preitl, Radu Emil Precup: Introduction to the Fuzzy Control, in Romanian: Introducere in conducerea fuzzy a proceselor, Editura Tehnica, Bucharest, 1997
- [9] Doru Todinca, Ana Maria Badulescu, Radu Emil Precup: VHDL Approach to Performance Analysis of Fuzzy Logic Controllers, in Proceedings of 8th Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference (IPMU 2000), Madrid, Spain, July 3-7, 2000, pp. 896-901
- [10] D. Todinca, R.-E. Precup: VHDL Simulation of a General Fuzzy Processor, in Proceedings of The International Conference on Technical Informatics CONTI'96, Timisoara, 1996, Vol. I, pp. 112-118
- [11] Masaki Togai, Hiroyuki Watanabe: Expert System on a Chip: An Engine for Real-Time Approximate Reasoning, in IEEE Expert Vol I, No 3, 1986, pp. 55-62