Expert Systems with Applications 41 (2014) 7046–7060

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Customizable hardware design of fuzzy controllers applied to autonomous car driving

Nadia Nedjah [a,*,1], Paulo Renato S.S. Sandres [a,1], Luiza de Macedo Mourelle [b,1]

[a] Department of Electronics Engineering and Telecommunication, Faculty of Engineering, State University of Rio de Janeiro, Brazil
[b] Department of Systems Engineering and Computation, Faculty of Engineering, State University of Rio de Janeiro, Brazil

ARTICLE INFO

Keywords:
Fuzzy logic control
Hardware architecture
Automatic car driving

ABSTRACT

Nowadays, final products often encompass a certain intelligence therein to deal with variation or lack of precision in the sensing input data. This intelligence is usually acquired via the utilization of existing soft techniques, such as artificial neural networks, genetic algorithms and fuzzy control, among others. Thus, it is profitable to have on-the-shelf shell scalable and adaptive hardware designs that implement these soft techniques. This availability allows for an immediate embedding of any of those designs onto final products. This usually entails a reduced time-to-market of the product. Process control is one of the many applications that took advantage of the fuzzy paradigm. In general, controllers are embedded into the controlled device. This paper presents a novel design of a reconfigurable efficient parallel architecture to implement fuzzy controllers on hardware with almost no design effort for final users. The proposed architecture is herein proven suitable for embedding. It is customizable, so it allows the setup and configuration of the controller parameters, and hence its use for any problem application. Two fuzzy controllers that model autonomous car driving are implemented and their cost and performance evaluated.

1. Introduction

Computational system modeling is full of ambiguous situations, wherein the designer cannot decide, with precision, what should be the outcome of the system. In Zadeh (1988), Zadeh introduced for the first time the concept of *fuzziness* as opposed to *crispiness* in data sets. When he invented *fuzzy sets* together with the underlying theory, Zadeh's main concern was to reduce system complexity and provide designers with a new computing paradigm that allows for approximate results. Whenever there is uncertainty, *fuzzy logic* together with *approximate reasoning* apply (Baldwin, 1981; Bandler & Kohout, 1981).

Fuzzy logic and approximate reasoning (Zadeh, 1984; Zadeh, 1988) can be used in applications ranging from system modeling and control to data clustering and prediction (Radecki, 1982). It has been used in many of applications, such as expert and learning fuzzy systems (Lin & Lee, 2011; Truong & Ahn, 2011; Yu, 2009; Zhang & Knoll, 1999), computing with words, approximate reasoning, natural language (Esragh & Mamdani, 1981), process control, robotics (Erdem, 2011; Ghidary, Hattori, Tadokoro, & Takamori, 2001; Petković, Issa, Pavlović, & Zentner, 2013), temperature and pressure control (Magdalena & Velasco, 1996), pattern recognition (Köppen & Nickolay, 2000; Melin & Castillo, 2013; Ozturk, Arslan, & Hardalac, 2008), financial decision making (Diao, Hellerstein, & Parekh, 2002) and data clustering (Nedjah & Mourelle, 2005), to name only few related applications. In spite of the fact that fuzzy Logic is a subject of great interest in the scientific and research circles, it is still not commonly used in industry, as it should be. Eventually, we found some literature containing practical equipment that are being currently used in industry (Mamdani & Pappis, 1977; Poorani, Urmila Priya, Udaya, & Renganarayanan, 2005).

Soft techniques are nowadays used to engineer final products that encompass certain intelligence within them. This intelligence is usually acquired via the use of artificial neural networks, genetic algorithms, evolutionary computation, fuzzy approximate reasoning, among others. It is profitable to have on-the-shelf shell scalable and adaptive hardware designs that implement these soft techniques (Chen & Hwang, 2009; Nedjah & Mourelle, 2005; Nedjah, Silva, & Mourelle, 2012). This allows for an immediate embedding onto the intended final product. This usually entails only a reduced time-to-market of the product. The purpose and motivation behind this work, which allows for the development of a reconfigurable hardware of a *shell* fuzzy controller, that can be reconfigured to the specification of the fuzzy logic model of any control application on-the-fly. The availability of the design

---

* Corresponding author. Tel.: +55 2198393142.
E-mail addresses: nadia@eng.uerj.br (N. Nedjah), psandres@globo.com (P.R.S.S. Sandres), ldmm@eng.uerj.br (L. de Macedo Mourelle).

[1] The authors are grateful to FAPERJ, CNPq and CAPES for their financial support.

http://dx.doi.org/10.1016/j.eswa.2014.05.032
0957-4174/

intellectual property allows to easily creating devices that take advantage of fuzzy logic control. A more optimistic objective is that, perhaps this would help spread the use of fuzzy logic control in final industrial products, such as planes and ships, among others.

In general, a hardware implementation can be either analog or digital based. The use of pure analog designs eliminates the need for any data conversion and thus preserves the precision dictated by the input data. One of the main implementation of fuzzy logic controllers using analog circuits can be found in Elmas, Deperlioglu, and Sayan (2009). Furthermore, the non-linear characteristics that are intrinsic to analog basic components always facilitate the implementation of the non-linear behavior of functions as needed by the fuzzy logic membership functions. However, analog circuits reduce a great deal the programmability of any design. On the other hand, digital design will certainly require conversion of input data and output result from analog to digital the other way around, respectively. These conversions introduce some quantization errors. Also, the implementation of non-linear behaviors are very time consuming when digital circuits are used. Nonetheless, digital circuits provide very high design programmability. In this paper, the proposed design is digital. The main motivation behind this work is to provide a customizable fuzzy controller design that can be reconfigured *on-the-fly* to fit the requirements of any application fuzzy model. We take advantage of simple precision floating-point data representation to minimize the loss of precision without occasioning a big impact on the controller response time.

There are many related works that implemented fuzzy controllers on FPGAs, as it will be detailed in Section 3. However, most of the reported works present designs for fuzzy logic controllers that are tailored for a specific application (Daijin, 2002; McKenna & Wilamowski, 2001; Poorani et al., 2005), depending, for instance on the number of input and output variables of the application as well as the number of linguistic terms used to defined each of these variables. The obtained design cannot be re-used to control another application even if there is only a tiny modification in the application, such as, for instance, a different number of linguistic terms in one of the variable fuzzy model. In contrast, the proposed hardware architecture is customizable. Any fuzzy controller can be accommodated. Taking advantage of the proposed intellectual property, the user can simply inform the required configuration of the fuzzy logic controller and synthesize the hardware specification model to yield the suitable fuzzy logic controller. The proposed design is fully customizable the user requirements in terms of number of input and output variable, the number of linguistic terms used to model them, the number and shape of rules that may be used during the inference process.

Furthermore, the reported designs do not use floating-point data representation. In most of system control applications, the floating-point data precision is crucial to the adequate sensibility of the controller design. In contrast, all the required computation in the proposed controller architectures is performed using simple-precision floating-point operations. Unlike these existing designs, the hardware design proposed herein exploits floating-point units to execute and deliver precise final results. This bestows more precision on the final outcome of the controller. Nonetheless, it imposes a higher hardware cost.

Existing hardware designs for fuzzy controllers are sequential. The fuzzification and defuzzification processes are scheduled to be executed sequentially. This delays the delivery of the control outputs and thus deteriorating the final quality of the control system. In the proposed design, we perform all input variable fuzzification simultaneously. Furthermore, all output variable defuzzification processes are also executed in parallel. This impacts positively the response time of the controller, thus improving tremendously the control quality of the system. The impact of this parallelism increases with the number of input and output variables of the application. Nevertheless, the hardware area requirement of the design due to this property also increases with the number of input and output variables. However, the trade-off is acceptable as the hardware resources available in the used FPGA are abundant and dedicated to the controller.

As far as we know, the proposed architecture is novel as there is no such intellectual property design available. This makes this work very significant to engineering new intelligent electronics devices based on fuzzy logic control. Any system or device wherein fuzzy control is used while performance in terms of response time and result precision is required, the proposed design would be suitable and applicable with no extra effort.

Summing up, the contributions of this paper are threefold. These are listed below according to their importance to designers and impact on the performance and cost of the final product:

(1) The proposed architecture is customizable so the hardware of any fuzzy logic controller model can be obtained for free, requiring only the fuzzy model as well as a hardware synthesis step using any tool that accepts VHDL, which is the hardware specification language used.
(2) It is precise as it exploits floating-point functional unit. Even though this decision has an impact on the response time and cost of the final product, it is worth it as the overall control quality improves a great deal.
(3) It is massively parallel as data fuzzification is done in parallel as well as defuzzification. This compensates on the response time but requires considerable hardware are requirements. Nonetheless, this can be sustained as any state-of-the-art FPGA provides abundant hardware resources.

Nonetheless, the proposed design does not give support all membership function possible shapes. It only permits the triangular model. Also, even though the number of linguistic fuzzy terms allowed can be parameterized in the current design, this is fixed for all variables. The architecture uses the number of linguistic terms of the most detailed variable. Note that this constraint yields a slightly higher area and time requirements than it is necessary when the number of linguistic terms is precisely tailored for the model.

The rest of this paper is organized into six sections. First, in Section 2, we briefly introduce some basic concepts of fuzzy logic controllers, which will be useful to follow on the description of the proposed architecture. After that, in Section 3, we go through some existing hardware designs for fuzzy logic controllers, reporting on their main features and explaining how these compare to the architecture proposed herein. Subsequently, in Section 4, we thoroughly describe the proposed customizable macro-architecture of the fuzzy controller. After that, in Section 5, we detail the micro-architecture of the main components of that macro-architecture. Then, in Section 6, we sketch the fuzzy model used for autonomous car driving, as reported in Li, Chang, and Chen (2003). Subsequently, in Section 7, we show, via the project the two fuzzy controllers, which model the process of autonomous car driving, that the proposed architecture is functionally operational, fully customizable and yet promising in terms of cost and performance. Finally, in Section 8, we draw some conclusions and point out some new directions for future work.

2. Basics of fuzzy control

Fuzzy control, which directly uses fuzzy rules, is the most important and common application of the fuzzy theory (Zadeh, 1968). Any fuzzy logic controller subsumes that each input and