Applied Soft Computing 13 (2013) 496–508

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

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

Particle swarm optimization of interval type-2 fuzzy systems for FPGA applications

Yazmin Maldonado, Oscar Castillo*, Patricia Melin

Tijuana Institute of Technology, Mexico

ARTICLE INFO

Article history:
Received 29 November 2011
Received in revised form 23 May 2012
Accepted 9 August 2012
Available online 21 August 2012

Keywords:
GA
PSO
AT2-FIS
AT2-FLC
FPGA
VHDL
XSG

ABSTRACT

This paper proposes the optimization of the type-2 membership functions for the average approximation of an interval of type-2 fuzzy controller (AT2-FLC) using PSO, where the optimization only considers certain points of the membership functions and, the fuzzy rules are not modified so that the algorithm minimizes the runtime. The AT2-FLC regulates the speed of a DC motor and is coded in VHDL for a FPGA Xilinx Spartan 3A. We compared the results of the optimization using PSO method with a genetic algorithm optimization of an AT2-FLC under uncertainty and the results are discussed. The main contribution of the paper is the design, simulation and implementation of PSO optimization of interval tye-2 fuzzy controllers for FPGA applications.

1. Introduction

Type-2 fuzzy logic systems (T2-FIS) are used successfully in many application areas, and these include control, classification, etc. [2,5,16], and there is an increasing interest in the research and implementations of T2-FIS because they offer more advantages in handling uncertainty with respect to type-1 fuzzy systems [20].

Fuzzy logic systems are based on rules, and these rules incorporate linguistic variables, linguistic terms and fuzzy rules. The acquisition of these rules is not an easy task for the expert and it is of vital importance in the operation of the controller. The process of adjusting these linguistic terms and rules is usually done by trial and error, which implies a difficult task, and for this reason there have been methods proposed to optimize those elements that over time have taken importance, such as genetic algorithms and particle swarm optimization [8,10].

Most of the fuzzy logic applications with physical systems require a real time operation, and the simple way to implement these systems is to realize them as software programs on general purpose computers, but this way cannot always be considered as a suitable design solution. Higher density programmable logic devices, such as the field programmable gate array (FPGA) can be used to integrate large amounts of logic in a single integrated circuit [3,21].

The research of different optimization techniques for type-2 fuzzy systems has increased, however there is the problem of runtime, and the runtime decreases when the implementation is processed in parallel, as in the FPGA. There are some works related to the optimization of a particular problem. In [7], they focus on two schemes: one is the memory hierarchy and the other is the algorithm design. Both the cache properties and the cache-aware development are investigated. To verify the effectiveness of the guidelines proposed, optimization techniques, including particle swarm optimization and the genetic algorithm, are employed. In [15], they describe the use of a genetic algorithm for the optimization of the parameters of a resonant controller employed for the control of current in a shunt active filter. The optimization process takes into account variations of the power system parameters to provide an optimal solution in terms of both speed of response and robustness. In [17] they propose the design of a type-2 fuzzy logic controller using GAs. The AT2-FLC was synthesized in VHDL code for FPGA, using XSG of Xilinx ISE and Matlab-Simulink. In [23] a classification is performed in given images into a discrete set of color classes, where the aim is to produce a fuzzy system for color classification and image segmentation with the least number of rules and minimum error rate using PSO method. There are optimization algorithms for FPGA implementation as in [27], they report the design of an IP core that implements a general-purpose GA engine that addresses these problems. It has been successfully synthesized

* Corresponding author.
E-mail address: ocastillo@hafsamx.org (O. Castillo).

1568-4946/$ – see front matter
http://dx.doi.org/10.1016/j.asoc.2012.08.032

and verified on a Xilinx Virtex II Pro field programmable gate arrays device.

This paper explains the design, simulation and implementation of T2-MFs optimization of the AT2-FLC for speed regulation of a DC motor (ReSDCM) in FPGA, based on an average approximation of intervals of type-2 fuzzy systems method [25], and the main goal of this paper is to compare the results (average errors and runtime) optimization of the T2-MFs using the PSO method and GA optimization. The proposed methodology for this paper is to synthesize the AT2-FLC in FPGA, where the optimization of the T2-MF takes place outside the FPGA, i.e. on a PC via serial port, and the optimized parameters of the T2-MFs are sent to the FPGA, this with the idea that once the AT2-FLC was optimized, the optimization process is disconnected from the PC and the AT2-FLC is ready for use. Fig. 1 shows the methodology diagram used for the optimization of T2-MFs for AT2-FLC in the FPGA.

For the optimization of triangular and trapezoidal type-2 membership functions we only consider some points of the T2-MFs for modification, where the fuzzy rules are not changed, and these with the aim of minimizing the runtime. To evaluate the ability of the best type-2 controller we consider three objective functions, the overshoot, undershoot and steady state error.

This paper has been organized as follows. In Section 2 we present an introduction of type-2 FIS, the PSO method, GA optimization and the FPGA device, in Section 3 we present the hardware design of AT2-FLC in VHDL for FPGA. The design of the PSO method and GA optimization for the AT2-FLC for ReSDCM are shown in Section 4, the T2-MF optimization results for AT2-FLC are shown in Section 5. In Section 5.1 we show the results for T2-MF with the PSO method using AT2-FLC in XSG, the results for T2-MF with the PSO for AT2-FLC using FPGA are given in Section 5.2, in Section 5.3 we present the results for T2-MF with the GA optimization using AT2-FLC in XSG, and results for T2-MF GA optimization using AT2-FLC in FPGA are given in Section 5.4. In Section 5.5 a comparison between PSO and GA optimization for AT2-FLC with uncertainty for XSG is presented. The comparison between PSO and GA optimization for AT2-FLC with uncertainty using FPGA are given in Section 5.6. Finally, Section 6 offers conclusions about this work.

2. Preliminaries

Fuzzy systems are being used more frequently, because they tolerate imprecise information and can be used to model nonlinear functions of arbitrary complexity. A fuzzy inference system (FIS) consists of three stages: fuzzification, inference and defuzzification [29]. Type-1 fuzzy systems (FIS-T1) have exact membership functions (MFs), while interval type-2 fuzzy systems (IT2-FIS) are described by membership functions with uncertainty [13,20].

2.1. Interval type-2 fuzzy inference system

The IT2-FIS consists of four stages: fuzzification, inference, type reduction and defuzzification. We describe below these stages.

2.1.1. Fuzzification

The fuzzification maps a numeric value $x = (x_1, \ldots, x_p)^T \in X_1 \times X_2 \times \cdots \times X_p \equiv X$, into a type-2 fuzzy set $\tilde{A}_x$ in X. $\tilde{A}_x$ is a singleton fuzzy set if $\mu_{\tilde{A}_x}(x) = 1/1$ for $x = x'$ and $\mu_{\tilde{A}_x}(x) = 1/0$ for all others $x \neq x'$ [19].

2.1.2. Inference

Fuzzy reasoning is performed using the rules and inference engine, it works the same way as for type-1 fuzzy systems, and except that the antecedents fuzzy sets and the consequent are represented by type-2 fuzzy sets. The process consists of combining the rules and mapping the input to the output (interval type-2 fuzzy sets), using the Join and Meet operations [19]. For an IT2-FIS with p inputs $x_1 \in X_1, x_2 \in X_2, \ldots, x_p \in X_p$ and one output $y \in Y$, it is assumed that there are M rules, and the lth rule in an IT2-FIS can be written as:

$$R^1: \quad \text{If } x_1 \text{ is } \tilde{F}_1^l \text{ and } x_p \text{ is } \tilde{F}_p^l, \quad \text{Then } y \text{ is } \tilde{G}^l \tag{1}$$

where $l = 1, \ldots, M$. Once we have the rules it is necessary to calculate the operations Join (⊔) and Meet (⊓) as well as sup-star composition (⋆) [19].

Fig. 2 shows the inference stage, x_1 and x_2 are the inputs, the inference stage uses the minimum t-norm, for the IT2-FIS the result

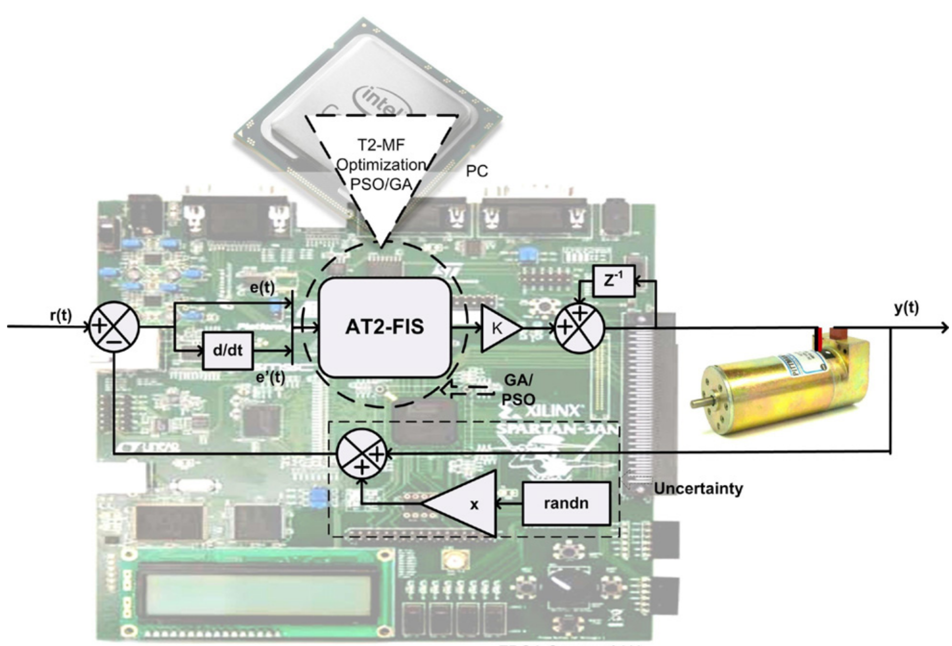

Fig. 1. Methodology used for the optimization of T2-MF for AT2-FLC in FPGA.