Applied Soft Computing 24 (2014) 1164–1174

Contents lists available at ScienceDirect

Applied Soft Computing

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

A multi-objective optimization of type-2 fuzzy control speed in FPGAs

Yazmin Maldonado, Oscar Castillo*, Patricia Melin

Tijuana Institute of Technology, Calzada Tecnologico s/n, 22379 Tijuana, Mexico

ARTICLE INFO

Article history:
Received 16 June 2012
Received in revised form 24 April 2014
Accepted 24 April 2014
Available online 2 July 2014

Keywords:
Multiobjective
GA
AT2-FLC
T1-FLC
ReSDCM
FPGA

ABSTRACT

This paper proposes a design of multi-objective genetic optimization for the average approximation of an interval type-2 fuzzy logic controller (AT2-FLC) using linear combination of individual criteria. There has been shown, on the real case, that the proposed method can be competitive to other commonly used approaches in the design of fuzzy controllers. The AT2-FLC was tested with different levels of uncertainty and different number of bits for the VHDL codification to regulate speed in a direct current motor (ReSDCM). Comparisons were made between the type-1 fuzzy logic controller versus AT2-FLC synthesized in VHDL code for FPGA and the AT2-FLC in VHDL versus the PID controller to ReSDCM. The main contribution of the paper is the design, implementation and comparison of multi-objective GA optimization of AT2-FLC for FPGA real applications. The AT2-FLC is targeted to a Xilinx Spartan 3A XC3S700A device using the Xilinx Foundation Environment.

Introduction and related work

Fuzzy controllers are commonly used due to their advantages in managing uncertainty in non-linear plants. There are at least two approaches for designing such controllers, leading to: type-1 (T1-FIS) and type-2 fuzzy inference system (IT2-FIS, AT2-FIS). Typically, a T1-FIS consists of three stages: fuzzification, inference and defuzzification [24]. A T1-FIS uses exact membership functions, while interval type-2 fuzzy inference systems (IT2-FIS) are described by membership functions with uncertainty [16,25]. The IT2-FIS consists of four stages: fuzzification, inference, type reduction and defuzzification [17]. The average approximation of interval type-2 fuzzy inference system (AT2-FIS) is an interval. Type-2 controllers have been designed as fuzzy inference system or average approximation of interval inference system. AT2-FIS is an interval type-2 fuzzy system, that involves replacing a type 2 fuzzy system using the average of two appropiate T1-FISs, and this method [23] is applied as follows: (1) replace each T2-MFs with two T1-MFs, with different degrees of membership in order to obtain the footprint of uncertainty. (2) In the type-2 inference stage it is necessary to compute the inference stage from each T1-FIS. (3) Compute the defuzzification of each T1-FIS, the results of the two systems are averaged instead of the type reduction and defuzzification stage. The hardware implementation of AT2-FIS was recognized as competitive to software implementation because offers significantly faster processing and ability of performing parallel processing [5].

An IT2-FIS and AT2-FIS can be implemented on a general purpose computer [1,2,9] or by a specific use of a microelectronics realization such as the FPGA. There are IT2-FISs implemented in computer software, but the required computer processing time particularly in real-time systems is slow. For this reason a FPGA hardware implementation of the AT2-FIS is proposed because the FPGA has fast processing and the ability of performing parallel processing [7].

The FPGA is a logic device that contains a two dimensional array of generic logic cells and programmable switches. A logic cell can be programmed to perform a simple function, and a programmable switch can be customized to provide interconnections among the logic cells. A custom design can be implemented by specifying the function of each logic cell and selectively setting the connection of each programmable switch. These cells contain IOBs, LUTs and slices. A logic cell contains a small configurable combinational circuit with a D-type flip-flop. The most common method to implement a configurable combinational circuit is a look-up table (LUT). An *n-input* LUT can be considered as a small $2n$ by 1 memory. The IOB is an input/output block of a Spartan 3A FPGA device provides a programmable interface between an I/O pin and the device's internal logic. It contains several storage registers and tri-state buffers as well as analog driver circuits that can be configured to provide different slew rates and driver strength and to support a variety of I/O standards. In Xilinx terms, two cells are grouped to form a slice, and four slices are grouped to form a configurable logic block (CLB) [4].

* Corresponding author. Tel.: +52 664 6236318.
E-mail addresses: ocastillo@tectijuana.mx, ocastillo@hafsamx.org (O. Castillo).

http://dx.doi.org/10.1016/j.asoc.2014.04.041
1568-4946/

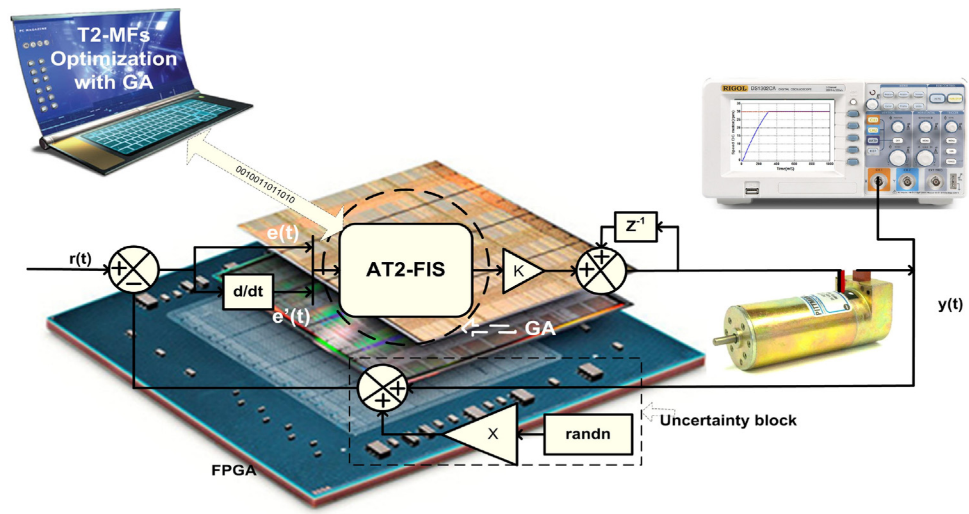

Fig. 1. Methodology used for genetic optimization of the AT2-FLC for ReSDCM.

The AT2-FIS and IT2-FIS are based on rules, which incorporate linguistic variables, linguistic terms and fuzzy rules; the acquisition of these rules is not an easy task for the expert and 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 for the designer, but there are many optimization methods such as genetic algorithms (GA) [11,22] that can be used to find optimal fuzzy systems designs [8]. Optimization algorithms help in the controller design and the model identification, most of the time the objective function depends on several criteria, in these cases it is better to use multiple criteria approach [5,19], which provide reliability and robustness in difficult search environments [6].

There are some works related to the optimization of a particular problem. In [3], a review of the methods used in the design of interval type-2 fuzzy controllers is presented. The fundamental focus of the work is based on the basic reasons for optimizing type-2 fuzzy controllers for different areas of application. They consider the application of genetic algorithms, particle swarm optimization and ant colony optimization as three different paradigms that help in the design of optimal type-2 fuzzy controllers. In [12], the authors propose a neuro-fuzzy inference circuit that generates membership function and inference rules automatically in the learning process. In this circuit they uses membership functions, which are generated by only using NOT operations and bit shift operations and tune only the parameters of the consequent part to reduce the circuit scale. This circuit is designed by using the Verilog HDL hardware description language and realized on the FPGA. In [18] they carried out the optimization and implementation of a fuzzy logic controller used as a maximum power point tracker for a standalone photovoltaic power system. The near optimum design for membership functions and control rules were found simultaneously by genetic algorithms. The fuzzy logic controller was designed, as well the components of the photovoltaic control unit, were implemented efficiently on a Xilinx reconfigurable FPGA chip using VHDL. In [21] the authors proposed a GA IP core that can be customized in terms of the population size, number of generations, crossover and mutation rates, random number generator seed and the fitness function. This proposal has been successfully synthesized and verified on a Xilinx Virtex II pro FPGA device with a clock speed of 50 MHz. The GA core has been used as a search engine for real time adaptive healing, but can be tailored to any given application by programming the values of the desired GA parameters.

This paper proposes a method for genetic optimization of the triangular and trapezoidal type-2 membership functions (T2-MFs) of an AT2-FIS for hardware applications such as the FPGA. This method involves considering only certain points of the membership functions, the fuzzy rules are not changed in order to give greater efficiency to the algorithm for use in real time applications. The GA is synthesized in Matlab code. The GA has been tested in both the AT2-FLC and T1-FLC.

The main contributions of this paper are the optimization of the fuzzy controller embedded in electronic systems, such as an FPGA using the average of two type-1 fuzzy systems, and with this, an approximation of a type-2 fuzzy system with less execution time was obtained. Different resolutions for hardware encoding were shown; the result was interesting because after 14 bits the error does not decrease but the response time (speed DC motor) is slower. We evaluate the advantages of type-2 fuzzy systems in the uncertainty characteristics offered.

Comparisons were made between the T1-FLC versus AT2-FLC in VHDL code and AT2-FLC in VHDL code versus PID controller to ReSDCM, to evaluate the difference in performance of three types of controllers, using the *t*-Student statistical test. Fig. 1 shows the methodology diagram used for the optimization of T2-MFs for AT2-FIS in the FPGA.

This paper has been organized as follows: in *Problem statement* section the description of the control problem is given, in *Multiobjective optimization approach* section the multiobjective approach of the T1-MFs and T2-MFs for T1-FLC and AT2-FLC respectively are presented, a design of AT2-FLC for ReSDCM in VHDL for FPGA is presented in *Design of the AT2-FIS for FPGA* section, discussion of experimental results between the T1-FLC and AT2-FLC versus PID controller are shown in *Discussion of experimental results* section. Finally, *Conclusion* section offers conclusions about this work.

Problem statement

The problem statement is formulated as follows, regulation speed of the DC motor in real environment, to solve this problem arise the following questions, which control type is adequatë?, fuzzy control?, conventional control (PID for example)?, which is the best device for programming the control speed?, FPGA?, microcontroller?. The following answers are given: fuzzy systems are used because they can solve linear and nonlinear problems regardless of the plant model, are also able to solve problems when the inputs are uncertain. The fuzzy system may be implemented in a