Applied Soft Computing 32 (2015) 175–188

Contents lists available at ScienceDirect

Applied Soft Computing

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

Hardware implementation and performance comparison of interval type-2 fuzzy logic controllers for real-time applications

Matthew D. Schrieber, Mohammad Biglarbegian*

School of Engineering, College of Physical and Engineering Science, University of Guelph, Guelph, ON N1G 2W1, Canada

ARTICLE INFO

Article history:
Received 25 August 2014
Received in revised form 10 March 2015
Accepted 11 March 2015
Available online 20 March 2015

Keywords:
Interval type-2 fuzzy logic
FPGA
Real-time applications

ABSTRACT

Due to the complex structure of IT2 FLCs, using them in real-time applications might be computationally expensive. To facilitate real-time implementation of these controllers, hardware with parallel processing abilities are recommended; field-programmable gate arrays (FPGA) are one class of such hardware. In this paper, we design and implement three different inference mechanisms of IT2 FLCs on hardware. These engines include Karnik–Mendel (KM) algorithms, Wu–Mendel (WM) uncertainty bounds, Nie-Tan (NT) and Biglarbegian–Melek–Mendel (BMM) which have recently been introduced in the literature. We first demonstrate how the proposed structures of the IT2 FLCs can be implemented on software; next, we propose architectures for implementing these IT2 FLCs on hardware. We performed simulations to compare the performance of the IT2 FLCs. To assess the controllers performance in real-time we used four indicators; the number of DSP48A1s, MUXCYs, slice registers and slice LUTs. It was shown that the NT and BMM controllers require significantly fewer resources compared to the other engines. While the controller that uses KM as its inference engine uses fewer resources in terms of DSP48A1s, it consumes a considerable amount of other resources compared to the WM controller. Finally, the transient responses of the controllers in terms of rise time and settling time were compared. It was found that the controllers with NT and BMM inference engines have faster closed-loop response in comparison to the one using the WM and KM. The results presented herein provides researchers and engineers a better insight into designing the most suitable IT2 FLCs, and hence it is expected IT2 FLCs can be implemented on hardware to further enable applications on plants requiring fast response (or ultimately real-time implementation).

1. Introduction

Recently, there has been a significant interest in using interval type-2 FLCs [1,2] largely because of their potential in handling uncertainty. A unique feature of IT2 FLCs lies in their upper and lower Figure function structure – called footprint of uncertainty – allowing them to accommodate potential uncertainties. IT2 FLCs also introduce the need for a type reducer which reduces the IT2 output to a type-1 (T1) type-reduced set. Many researchers have concluded that IT2 FLCs are capable of outperforming some traditional controllers as well as T1 FLCs in several applications [3,4]. For example, Chen and Tan [5] proposed a T2 FLC that not only was proven to be stable through Lyapunov analysis, but was also found to be more robust and perform better than an adaptive T1 counterpart. Despite IT2 FLCs being more capable of handling uncertainty in comparison to T1, their complex structure has limited their ability to be utilized on a larger scale.

The complex structure of IT2 FLCs compared to T1 FLCs can be attributed to the type-reduction process. The well-known Karnik–Mendel (KM) algorithms computes the type-reduced set at a high computational cost [6]. As such, using the KM algorithms is significantly more challenging than other IT2 FLC mechanisms due to the iterative nature of these algorithms [7,8]. Karimi and Safarinejadian [4] proposed a T2 FLC for control of a simple nonlinear system, but experienced problems when using the controller on-line due to computationally expensive operations of their proposed T2 FLC. The computational complexity of any IT2 FLC invariably leads to a significant increase in processing time, especially for applications required to operate at high speeds. As such, researchers are attempting to propose simpler structures for such controllers.

Substantial research on improving and optimizing IT2 FLCs is currently being conducted. For example, Juang and Hsu [9] designed an IT2 FLC for a mobile robot in a wall-following application capable of operating without any *a priori* rule sets using

* Corresponding author. Tel.: +1 5198244120x56248.
E-mail addresses: schriebe@uoguelph.ca (M.D. Schrieber), mbiglarb@uoguelph.ca (M. Biglarbegian).

http://dx.doi.org/10.1016/j.asoc.2015.03.022
1568-4946/

the ant colony optimization technique to generate rules automatically. In [10,11] the use of Look-Up Tables (LUTs) was proposed to facilitate the storage of pre-calculated type-reduced set values. However, this structure can only function when a set of constants are used for rule consequent values, and does not work for Takagi–Sugeno–Kang (TSK) method where the rule consequent values are dependent on the input values. Lycnh et al. have successfully implemented both a KM and Wu–Mendel (WM) IT2 FLC using Mamdani model structure and found that the WM controller uses an average of 45% less clock cycles than the KM controller [12]. More recent research into new type-reduction techniques beyond KM and WM have shown significant improvements in terms of real-time performance. In [13], a unified output process is introduced that encompasses both type-reduction and defuzzification. The new process was shown to be capable of controlling nonlinear plants such as ball and beam and inverted pendulum nonlinear plants, achieving a settling time of 1.43 s and 0.74 s, respectively.

Researchers have begun exploring the implementation of IT2 FLCs on hardware to increase the speed of the control process [7,14,15]. In [16], genetic algorithms were used for the optimum design of membership functions (triangular and trapezoidal) of a fuzzy system to be implemented in hardware such as FPGA. Simulations were performed using Matlab and VHDL code and control performance of type-1 and type-2 fuzzy controllers were made. In [17], particle swarm optimization method was used for optimum design of IT2 FLCs and was coded in VHDL for a Xilinx Spartan FPGA. The results were compared with the IT2 FLC designed using genetic algorithms as well. Most recently, Maldonado et al. in [18] and [19] use particle swarm and multi-objective genetic algorithm optimization methods for the design of an IT2 FLC in FPGA applications. In another work [20], an IT2 FLC was developed for FPGA to be implementation on a DC motor to regulate it speed. Melgarejo et al. [21] implemented a WM IT2 FLC on a field programmable gate array (FPGA) to take advantage of the parallelism that this hardware offers. The developed controller was implemented successfully with a papered operating speed of 33.3 ns. However, this work was not used in a closed-loop system that includes a plant in which the system performance would have been significantly affected by the controller operating over a papered 9 clock cycles. Huang and Tsai [15] applied their controller to an autonomous omnidirectional mobile robot and achieved similar results to that of [21]. The need for robustness and stability in developing the controller for the omnidirectional mobile robot platform [15] increases the complexity of the controller which makes it difficult for implementation on hardware, and hence a hardware/software co-design was employed instead. Because of the hardware/software co-design, the controller was only able to achieve an operating speed of 150 μs, approximately 4500 times slower than what was papered in [21]. Recently, the concept of hardware implementation of IT2 FLCs was further explored in [7] where an IT2 FLC with Mamdani model structure was designed for hardware implementation. For speed control of a DC Motor simulation results showed the desired controller performance in which the potential operating speed in comparison to a general-purpose industrial computer featuring a quad-core processor was papered as 225,000–450,000 times faster. However, this potential speed-up is based on a 5 clock cycle process which will most likely affect the system performance in closed-loop. In another work, an IT2 FLC with Mamdani structure for a mobile robot was implemented on an FPGA [14]; however, any indication of real-time performance of the controller was not given. In another work, an entirely new type-reducer is introduced in [22] where it is used in conjunction with a T2 neural fuzzy system learned through its T1 counterpart in a hardware based implementation. Due to the type-reducer not requiring a division operation the controller was capable of operating at a much faster rate than that of anything else being papered in current literature, even when applied to nonlinear plants. This paper presented an FPGA based controller featuring two inputs and 6 rules with an 8-bit resolution and was capable of a 4 stage pipeline that ran at a clock rate of 85.668 MHz, which yielded a total runtime from input to output of only 46.69 ns.

In this paper, we present a hardware implementation for several IT2 FLCs by considering the specific hardware limitations with regards to device area and real-time performance. Unlike existing approaches that have used Mamdani model structure, we use the TSK model structure for implementing the controllers because of its closed-form mathematical structure which makes it more suitable for fast control applications. We design and implement three different inference mechanisms of IT2 FLCs in software; next, we propose architecture for implementing these IT2 FLCs on hardware. These engines include KM, WM and NT methods, as well as BMM [1,23] which has recently been introduced in the literature. We first propose a structure for implementing the BMM engines on FPGA. Unlike previous papers in the literature, we propose details on how to implement IT2 FLCs on hardware easily which enables controls engineer with some FPGA background the tools required for implementing any IT2 FLC in hardware. To the best of our knowledge this is the first paper in the literature that compares the real-time performances of several IT2 FLC inference engines in terms of transient response as well as resource allocations. We believe this is an original work that tries to compare the performance of different IT2 FLCs in these aspects. As such, it is of interest to control engineers and practitioners. We also provide a new division algorithm which enables implementation of these engines on FPGA. This has not been discussed or addressed in any of the previous work in the literature. In this paper, we introduce four new indicators to compute the consumed resources by these engines when they are implemented on FPGA. These indicators to have meaningful metrics for designers and practitioners when implementing IT2 FLCs on hardware such as FPGAs."

To the best of our knowledge, there is no published work in the literature to provide a real-time comparison amongst these engines either. In this paper, the contributions are summarized as follows: (1) We develop hardware architectures for the above four inference engines on FPGA; (2) provide metrics or indicators to measure their real-time performance; (3) demonstrate the developed methodologies and indicators for several plants.

The organization of the rest of this paper is as follows: Section 2 contains general background on IT2 TSK FLCs and hardware design. Section 3 presents the controller design used in the hardware implementation. Section 4 demonstrates the hardware implementation of the IT2 FLC components. Section 5 provides simulation and experimental results. Finally, Section 6 concludes the paper.

2. Background

This section contains background on FLCs and hardware design. First, background T2 fuzzy logic systems is presented. Next, an overview of IT2 TSK models is provided. Finally, a detailed overview is given for hardware design and field programmable gate array (FPGA) devices.

2.1. Interval type-2 FLCs

Interval type-2 (IT2) FLCs were developed as a mean to incorporate more uncertainty. IT2 FLCs are capable of handling uncertainty at the MF level, which allows designers a greater degree of freedom in designing the MFs. IT2 MFs feature two T1 MFs in the form of an upper (*UMF*) and a lower (*LMF*) MF that bound a footprint of uncertainty (*FOU*) as shown in Fig. 1.