ISA Transactions 53 (2014) 1583–1591

Contents lists available at ScienceDirect

ISA Transactions

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

A controller based on Optimal Type-2 Fuzzy Logic: Systematic design, optimization and real-time implementation

H.M. Fayek [a], I. Elamvazuthi [a,*], N. Perumal [a], B. Venkatesh [b]

[a] Department of Electrical and Electronic Engineering, Universiti Teknologi PETRONAS, 31750 Tronoh, Perak, Malaysia
[b] Department of Electrical and Computer Engineering, Ryerson University, Toronto, Ontario, Canada M5B 2K3

ARTICLE INFO

Article history:
Received 11 April 2014
Received in revised form
21 May 2014
Accepted 2 June 2014
Available online 21 June 2014
This paper was recommended for publication by Dr. Jeff Pieper

Keywords:
Intelligent control
Type-2 Fuzzy Logic
Fuzzy Logic Control
Optimization
Robotic arm

ABSTRACT

A computationally-efficient systematic procedure to design an Optimal Type-2 Fuzzy Logic Controller (OT2FLC) is proposed. The main scheme is to optimize the gains of the controller using Particle Swarm Optimization (PSO), then optimize only two parameters per type-2 membership function using Genetic Algorithm (GA). The proposed OT2FLC was implemented in real-time to control the position of a DC servomotor, which is part of a robotic arm. The performance judgments were carried out based on the Integral Absolute Error (IAE), as well as the computational cost. Various type-2 defuzzification methods were investigated in real-time. A comparative analysis with an Optimal Type-1 Fuzzy Logic Controller (OT1FLC) and a PI controller, demonstrated OT2FLC's superiority; which is evident in handling uncertainty and imprecision induced in the system by means of noise and disturbances.

1. Introduction

Fuzzy Logic Control (FLC) has demonstrated superiority over classical control, namely Proportional-Integral-Derivate (PID) controllers, especially in applications where imprecision and uncertainty are present in the system [1]. Generally, FLC has proven to be more superior in terms of (1) noise rejection, (2) flexibility, (3) the use of human knowledge, not accurate mathematical models, and (4) sensitivity to disturbances, which finally yields (5) overall better system performance [2]. Type-2 Fuzzy Logic (T2FL) was introduced to generalize Type-1 Fuzzy Logic (T1FL) [3–4]. T2FL is able to model uncertainty and imprecision in a much better way [5], which makes Type-2 Fuzzy Logic Control (T2FLC) ideal for control applications. However, T2FL is more difficult to understand and implement than the conventional T1FL [6]. The lack of systematic design procedures for T2FL controllers, as well as T1FL controllers, has been a challenge to researchers and engineers and is considered as one of the drawbacks of T2FLC and FLC.

Various designs of T2FL controllers were reported in the literature, most of which were designed based on heuristic methods. For example, in [7], a T2FL controller was designed to control the speed of a DC motor and judged it against a T1FL controller, demonstrating that the T2FL controller had better performance. In [8], the design of an adaptive type-2 fuzzy-neuro system for controlling the position of a servo system with an intelligent sensor was presented. A comparative analysis between different types and numbers of type-2 membership functions and their impact on a T2FL controller's performance to control a servomotor was carried out in [9].

A systemic design procedure of a stable T2FL controller was presented in [10] based on fuzzy Lyapunov synthesis [11], where only the rules of the T2FL controller were considered in the systematic design process. Optimization algorithms such as Genetic Algorithm (GA) or Particle Swarm Optimization (PSO) are able to provide a systemic design procedure as reported in [12]. This optimization process is conducted off-line using a mathematical model of the desired plant, which may result in unsatisfactory results due to the differences between the mathematical model and the actual plant. This problem is dealt with in case of T2FL, since T2FL is able to deal with such imprecision outstandingly. However, optimization of T2FL introduces a new problem, which is the lengthy and computationally expensive optimization process due to the large number of points that need to be considered in the optimization process for each membership function as reported in [13], where the authors used GA to design the membership functions of a T2FL controller, and stated that the computation time needed for the optimization process was too lengthy. This problem was tackled in [14] by assigning fixed values

* Corresponding author. Department of Electrical and Electronic Engineering, Universiti Teknologi PETRONAS, Block 23, 31750 Tronoh, Perak, Malaysia. Tel.: +60 53687882.
E-mail addresses: haytham.fayek@ieee.org (H.M. Fayek), irraivan_elamvazuthi@petronas.com.my (I. Elamvazuthi).

http://dx.doi.org/10.1016/j.isatra.2014.06.001
0019-0578/

for some points of the membership functions and optimizing the remaining points using PSO or GA without degrading the performance of the controller.

A few papers presented the implementation of a T2FL controller in real-time. In [15], a non-optimal T2FL controller was designed and implemented to control an autonomous mobile robot demonstrating better results than the T1FL counterpart under similar conditions. In [16], the design – using heuristic methods – and real-time implementation of a T2FL controller with different defuzzification methods to control a DC Servomotor was presented and the computation time was recorded as well. It was demonstrated that although T2FL was computationally more expensive, it yielded improved handling of noise and system disturbances than the T1FL controller. In [14], the membership functions of a T2FL controller were optimized and the controller was implemented on a Field-Programmable Gate Array (FPGA) to control a motor with backlash, demonstrating better results in case of the T1FL controller as well.

This article presents a systematic design procedure of an Optimal Type-2 Fuzzy Logic Controller (OT2FLC), where both the gains and membership functions of the OT2FLC are considered in the optimization process. PSO was used to optimize the gains of the inputs and output of the controller. While, GA was used to optimize the membership functions by considering only two parameters for each type-2 membership function to decrease the computational complexity of the optimization process, while ensuring robustness of the optimal controller. The proposed controller is implemented in real-time to control the position of an actual DC servomotor, investigating various defuzzification methods. The performance and computation time of the OT2FLC are compared against an Optimal Type-1 Fuzzy Logic Controller (OT1FLC) and an optimal Proportional-Integral (PI) controller.

The remainder of this article is organized as follows: In Section 2, type-2 fuzzy sets and systems theories are presented, as well as a brief review on PSO and GA. The modeling, design methodology, optimization and implementation of the OT2FLC to control the position of the DC servomotor are explicated in Section 3. Section 4 depicts the real-time performance of the proposed OT2FLC and discusses the results against the OT1FLC and the conventional PI controller. The article is concluded in Section 5.

2. Proposed techniques

2.1. Type-2 Fuzzy Logic

Zadeh introduced Type-2 Fuzzy Sets (T2FS) as an extension to Type-1 Fuzzy Sets (T1FS) [17,18]. T2FS are able to handle uncertainty in a much better way, which make T2FS ideal for control applications. A T2FS, $\tilde{A}$ is characterized by a type-2 membership function $\mu_{\tilde{A}}(z, \mu(z))$, where $z \in Z$ and $\mu \in J_z \subseteq [0,1]$, as follows:

$$\tilde{A} = \{((z, \mu(z)), \mu_{\tilde{A}}(z, \mu(z)))| \forall z \in Z, \ \forall \mu(z) \in J_z \subseteq [0,1]\} \quad (1)$$

where, $0 \le \mu_{\tilde{A}}(z, \mu(z)) \le 1$, and Z is the universe of discourse. $\tilde{A}$ may also be represented as follows [19]:

$$\tilde{A} = \int_{z \in Z} \int_{\mu(z) \in J_z} \frac{\mu_{\tilde{A}}(z, \mu(z))}{(z, \mu(z))} \quad (2)$$

where, $J_z \subseteq [0,1]$, $\int\int$ resembles union over all admissible z and $\mu(z)$ [20] and J_z is called primary membership of z. Concretely, a type-2 membership function comprises an inferior membership function and a superior membership function; each function is represented by a type-1 membership function. The superior membership function is denoted as Upper Membership Function (UMF), while the inferior membership function is denoted as Lower Membership Function (LMF) [21] as in Fig. 1. The interval between these two membership functions represents the Footprint Of Uncertainty (FOU), which distinguishes the T2FS [22].

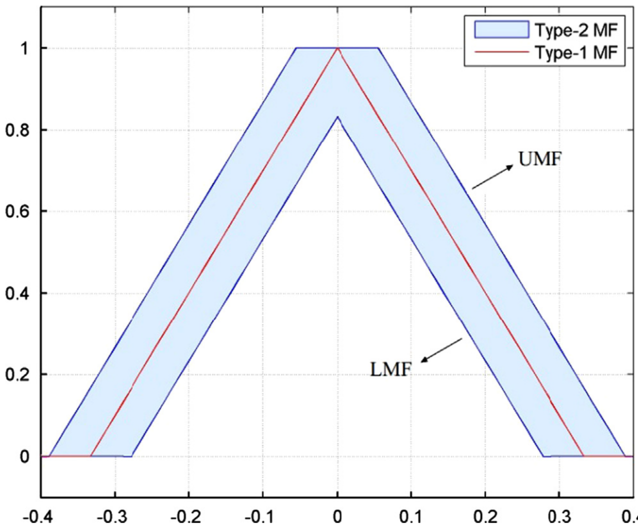

Fig. 1. Type-2 fuzzy membership function.

An Interval Type-2 Fuzzy Set (IT2FS) is the most widely used type of the T2FS [19]. It can be considered as a special case of T2FS, whereas an IT2FS is one in which the membership grade of every domain point is a crisp set whose domain is some interval contained in the interval [0,1]. The membership grade of an IT2FS is an interval set with a unity value for each secondary grade in that set [23]. The IT2FS was used in this work.

A Type-2 Fuzzy Inference System (T2FIS) has the same IF-THEN rules as the conventional type-1 Fuzzy Inference System (FIS) except that the antecedent and consequent are in type-2 form as follows:

$$R^l : \text{IF } x_1 \text{ is } \tilde{V}_1^l \text{ AND} \ldots x_n \text{ is } \tilde{V}_n^l \quad \text{THEN } y \text{ is } \tilde{W}_{l,} \quad l = 1, \ldots, M \quad (3)$$

where, $\tilde{V}_n^l$ is a type-2 antecedent, $y \in Y$ is the output, and $\tilde{W}_l$ is a type-2 consequent. The structure of a T2FL system, which is shown in Fig. 2, is very similar to a T1FL system. A T2FL system comprises a type-2 fuzzifier, a type-2 rule-base, a type-2 inference engine, and substitutes the type-1 defuzzifier with an output processor which includes a type reducer and a type-2 defuzzifier [24].

The Extension Principle [25] is used to extend each type-1 defuzzification method for the corresponding type-reduced set. A type-reducer combines the output sets in some way and then performs a centroid calculation on this T2FS, which leads to a T1FS that is called the type-reduced set. Type-2 defuzzification methods used in this paper are [25,26]

- ***Centroid***: The centroid type-reducer combines all the rule-output T2FS by finding their union as in Eq. (4).

$$\mu_{\tilde{B}}(y) = \cup_{l=1}^{M} \mu_{\tilde{B}^l}(y) \quad \forall y \in Y \quad (4)$$

where, $\mu_{\tilde{B}^l}$ is the secondary membership function for the lth rule. Finding the union of T2FS requires computing the join of their secondary membership functions. This method involves an enormous amount of computation, as the centroid and membership computations have to be repeated numerous times. Eq. (5) is used by the centroid type-reducer to calculate the centroid of $\tilde{B}$:

$$Y_c(x) = \int_{\theta_1 \in J_{y1}} \cdots \int_{\theta_N \in J_{yN}} [f_{y1}(\theta_1) \star \ldots \star f_{yN}(\theta_N)] \Bigg/ \frac{\sum_{i=1}^{N} y_i \theta_i}{\sum_{i=1}^{N} \theta_i} \quad (5)$$

where, $i = 1, \ldots, N$, and $\theta_N J_{yi} f_{yi}$ are associated with $\mu_{\tilde{B}}(y)$.

- *Center of sums:* The center of sums type-reducer combines the type-2 rule output sets by adding their secondary membership