Engineering Applications of Artificial Intelligence 42 (2015) 36–44

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Genetic-algorithm-based type reduction algorithm for interval type-2 fuzzy logic controllers

Tzyy-Chyang Lu

Advanced Institute of Manufacturing with High-Tech Innovations, National Chung Cheng University, Chiayi, Taiwan

ARTICLE INFO

Article history:
Received 27 October 2014
Received in revised form
6 January 2015
Accepted 23 February 2015
Available online 2 April 2015

Keywords:
Interval type-2 fuzzy logic controllers (IT2-FLCs)
Type reduction
Optimization

ABSTRACT

In interval type-2 fuzzy logic controllers (IT2-FLCs), the output processing includes type reduction and defuzzification. Recently, researchers have proposed many efficient type reduction algorithms, but there are no effective schemes to improve the output of defuzzification. This paper presents a genetic-algorithm-based type reduction algorithm, which reduces the type of an interval type-2 fuzzy set and provides optimal defuzzified output from the type-reduced set. In addition, the proposed type reduction is executed offline (in other words, the controller has been reduced to type-1 in practical applications), which significantly reduces the computational cost and facilitates the design of controllers that operate in real time. To demonstrate the effectiveness of the proposed method, truck backing control problems are utilized. The results show that the proposed method outperforms general IT2-FLCs in terms of speed, computational cost, and robustness.

1. Introduction

Fuzzy logic controllers (FLCs) have been successfully applied to a wide variety of applications. There have been some publications in the design of type-1 fuzzy logic controller (T1-FLC). For example, Bingul and Karahan (2011) and Bouallegue et al. (2012) employ the particle swarm optimization algorithm to tune the FLC structures, which applied to the robot trajectory control and the electrical DC drive speed control, respectively. Cazarez-Castro et al. (2012) use fuzzy Lyapunov synthesis to design the FLCs to solve the output regulation problem of a servomechanism with nonlinear backlash. Mendes et al. (2014) use a hierarchical genetic algorithm (GA) to automatically extract all fuzzy parameters of a FLC in order to control nonlinear industrial processes.

Interval type-2 fuzzy logic systems are an extension of traditional type-1 fuzzy logic systems. When the information is so fuzzy that even defining the membership function values in the interval [0,1] is difficult, a type-2 membership function is beneficial (Hosseini et al., 2012). A type-2 membership function can be considered as a collection of different embedded type-1 fuzzy sets, which construct the footprint of uncertainty (FOU) (Hu et al., 2012). Mendel (2001) and Hagras (2004) have shown that using type-2 fuzzy sets will result in the reduction of the rule base compared to that obtained using type-1 fuzzy sets. Furthermore, the extra degrees of freedom provided by the FOU enables an interval type-2 fuzzy logic controller (IT2-FLC) to produce outputs that cannot be achieved by a T1-FLC with the same number of membership functions. Thus, an IT2-FLC is able to model more complex input–output relationships than its type-1 counterpart and, thus, can give better control response (Coupland et al., 2006; Biglarbegian et al., 2009). Recently, there has been a growing interest in using IT2-FLC in many applications as well as in control processes (Biglarbegian et al., 2011; Hosseini et al., 2012; Cara et al., 2013).

The major difference between a T1-FLC and an IT2-FLC is that for the latter, at least one of the membership functions in the rule base is a type-2 membership function (Sepulveda et al., 2007). Hence, the inference engine outputs are type-2 fuzzy sets, and type reduction is needed to convert them into a type-reduced set (Wu, 2006, 2013). A type-reduced set is an interval type-1 set defined by the left and right centroids (Karnik and Mendel, 1999, 2001). The crisp output can be any value from this interval set depending on uncertainties (Ulu et al., 2011). Karnik–Mendel algorithms (KMAs) (Karnik and Mendel, 2001) are the most popular type reduction approach for computing the boundary centroids. However, they suffer from two major shortcomings. First, KMAs are iterative algorithms that use a trial-and-error process of testing various centroids to find the boundary ones for all input values. This is time-consuming and thus cannot be realized in real time. Second, in the defuzzification process, KMAs take the average of these boundary centroids as the defuzzified output, which may not be a good choice.

In recent years, some variants of type reduction algorithms have been proposed. These algorithms can be classified into two categories: enhanced KMAs (EKMAs) (Wu and Mendel, 2009; Yeh

E-mail address: tclu2012@gmail.com

http://dx.doi.org/10.1016/j.engappai.2015.02.012
0952-1976/

et al., 2011; Hu et al., 2010, 2012) and alternative type reduction algorithms (ATRAs) (Gorzalczany, 1987; Liang and Mendel, 2000; Wu and Mendel, 2002; Wu and Tan, 2005; Coupland and John, 2007; Greenfield et al., 2008; Nie and Tan, 2008; Du and Ying, 2010; Tao et al., 2012). EKMAs improve directly over the original KMAs, obtaining the same boundary centroids using fewer iterative computations. However, in the defuzzification process, they still take the average of the left and right centroids as a defuzzified output. In other words, EKMAs give the exact same outputs as those of the original KMAs but faster. Unlike EKMAs, ATRAs have closed-form representations, which give the approximate outputs from KMAs. These algorithms are usually much faster (but not necessarily have better control response) than the original KMAs. Ulu et al. (2011) proposed a dynamic defuzzification method that uses a linear combination of the boundary centroids to enhance the control response. However, since the boundary centroids are still computed using KMAs, the dynamic defuzzification method has the same computational load as that of the general defuzzification method.

Instead of taking the average of the left and right centroids, the present work employs GA to find the optimal defuzzified output from the type-reduced set. In addition, the GA-based type reduction is executed offline (in other words, the controller has been reduced to type-1 in practical applications), which significantly reduces the computational cost and facilitates the design of controllers that operate in real time. Finally, and most importantly, the designed controller is type-1 in practical applications but has the complex control surfaces like type-2. This means that the designed controller possesses more degrees of freedom in design aspects like a type-2 controller.

The rest of this paper is organized as follows. Section 2 briefly reviews IT2-FLCs. Section 3 describes the proposed genetic-algorithm-based type reduction. Section 4 shows the results of the proposed method applied to the truck backing up control problem. Finally, Section 5 gives the conclusions.

2. Interval type-2 fuzzy logic controllers

An interval type-2 fuzzy set (IT2-FS) $\tilde{A}$ is characterized by an interval type-2 membership function as (Wu, 2013)

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x \subseteq [0,1]} 1/(x,u) \tag{1}$$

where x is the primary variable in domain X, $u \in [0,1]$ is the secondary variable in domain J_x at each $x \in X$, and J_x is called the primary membership of x. The secondary grades of $\tilde{A}$ are all equal to 1.

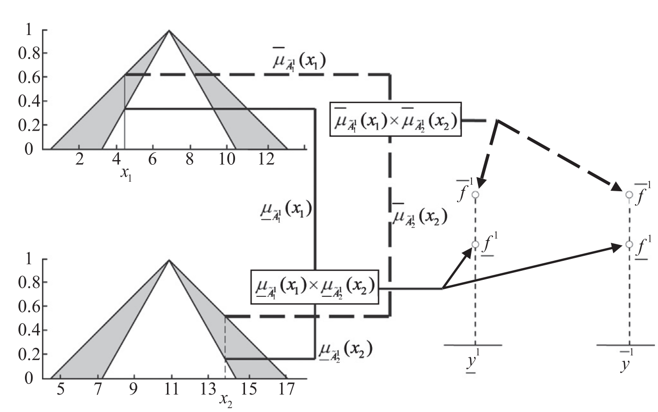

Fig. 1. Two interval type-2 membership functions of antecedent part for rule $\tilde{R}^1$ (left). Weighing interval sets and corresponding fired interval type-2 sets for first output (right).

Consider the rule base of an IT2-FLC composed of M rules with the following form:

$\tilde{R}^i$: IF x_1 is $\tilde{A}_1^i$ and … and x_n is $\tilde{A}_n^i$, THEN y is Y^i

where $i = 1, 2, \ldots, M$, $\tilde{A}_j^i$, $j = 1, \ldots, n$, is an IT2-FS, and $Y^i = [\underline{y}^i, \bar{y}^i]$ is an interval, which can be understood as the centroid of a consequent IT2-FS, or the simplest Takagi–Sugeno–Kang (TSK) model (Wu, 2013).

For an input vector $\boldsymbol{x} = x_j$, $j = 1, \ldots, n$, the membership degree $\mu_{\tilde{A}_j^i}(x_j)$ is an interval set, denoted by

$$\mu_{\tilde{A}_j^i}(x_j) = \left[\underline{\mu}_{\tilde{A}_j^i}(x_j), \bar{\mu}_{\tilde{A}_j^i}(x_j)\right] \tag{2}$$

where $i = 1, \ldots, M$, $j = 1, \ldots, n$. As only interval type-2 sets are used and the *meet* operation is implemented with the *product* t-norm, the firing set is the following type-1 interval set (Wu and Tan, 2006):

$$F^i(\boldsymbol{x}) = \left[\prod_{j=1}^{n} \underline{\mu}_{\tilde{A}_j^i}(x_j), \prod_{j=1}^{n} \bar{\mu}_{\tilde{A}_j^i}(x_j)\right] = \left[\underline{f}^i, \bar{f}^i\right] \tag{3}$$

where $i = 1, \ldots, M$. An example of two interval type-2 membership functions in the antecedent part for rule $\tilde{R}^1$ and the corresponding fired interval type-2 sets for the first output are shown in Fig. 1.

Type reduction is performed to combine $F^i(\boldsymbol{x})$ and the corresponding rule consequents. The center-of-sets type reduction (Karnik and Mendel, 2001; Shill et al., 2012) is used in this paper. Then, $C(\boldsymbol{x})$ is all possible combinations of the centroids:

$$C(\boldsymbol{x}) = \bigcup_{\substack{f^i \in F^i(\boldsymbol{x}) \\ y^i \in Y^i}} \frac{\Sigma_{i=1}^{M} f^i y^i}{\Sigma_{i=1}^{M} f^i} = [c_{\min},\ c_{\max}] \tag{4}$$

where

$$c_{\min} = \min_{L \in [1, M-1]} \frac{\sum_{i=1}^{L} \bar{f}^i \underline{y}^i + \sum_{i=L+1}^{M} \underline{f}^i \underline{y}^i}{\sum_{i=1}^{L} \bar{f}^i + \sum_{i=L+1}^{M} \underline{f}^i} \tag{5}$$

$$c_{\max} = \max_{R \in [1, M-1]} \frac{\sum_{i=1}^{R} \underline{f}^i \bar{y}^i + \sum_{i=R+1}^{M} \bar{f}^i \bar{y}^i}{\sum_{i=1}^{R} \underline{f}^i + \sum_{i=R+1}^{M} \bar{f}^i} \tag{6}$$

L and R are switch points, and $\left\{\underline{f}^i\right\}$ and $\left\{\bar{f}^i\right\}$ have been sorted in ascending order (Mendel, 2001; Mendel and Wu, 2010). L and R can be computed using KMAs or their variants. Take $c_{\min}$ of Fig. 2(a) as an example. An iterative procedure finds the switch point L satisfying:

$$\underline{y}^L \le c_{\min} < \underline{y}^{L+1} \tag{7}$$

For $i \le L$, the upper bounds of the firing intervals are used to calculate $c_{\min}$; for $i > L$, the lower bounds are used. This ensures that $c_{\min}$ is the minimum. A similar procedure can be used to find the switch point R to ensure that $c_{\max}$ is the maximum (see Fig. 2(b)). Finally, the crisp output can be obtained by taking the average of $c_{\min}$ and $c_{\max}$:

$$y = \frac{c_{\min} + c_{\max}}{2} \tag{8}$$

3. Genetic-algorithm-based type reduction

According to the above analysis, the defuzzified output y is determined by the boundary centroids in (4). However, taking the average of these centroids as a defuzzified output may not be a good choice. In this work, the boundary centroids are replaced by the