Applied Soft Computing xxx (2012) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

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

# A review on the design and optimization of interval type-2 fuzzy controllers

Oscar Castillo*, Patricia Melin

*Tijuana Institute of Technology Tijuana, Mexico*

ARTICLE INFO

*Article history:*
Received 17 August 2011
Received in revised form
13 December 2011
Accepted 18 December 2011
Available online xxx

*Keywords:*
Type-2 fuzzy controllers
Bio-inspired methods
Design and optimization

ABSTRACT

A review of the methods used in the design of interval type-2 fuzzy controllers has been considered in this work. The fundamental focus of the work is based on the basic reasons for optimizing type-2 fuzzy controllers for different areas of application. Recently, bio-inspired methods have emerged as powerful optimization algorithms for solving complex problems. In the case of designing type-2 fuzzy controllers for particular applications, the use of bio-inspired optimization methods have helped in the complex task of finding the appropriate parameter values and structure of the fuzzy systems. In this review, we 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. We also mention alternative approaches to designing type-2 fuzzy controllers without optimization techniques. We also provide a comparison of the different optimization methods for the case of designing type-2 fuzzy controllers.



## 1. Introduction

Uncertainty affects decision-making and emerges in a number of different forms. The concept of information is inherently associated with the concept of uncertainty [49,53]. The most fundamental aspect of this connection is that the uncertainty involved in any problem-solving situation is a result of some information deficiency, which may be incomplete, imprecise, fragmentary, not fully reliable, vague, contradictory, or deficient in some other way. Uncertainty is an attribute of information [69]. The general framework of fuzzy reasoning allows handling much of this uncertainty and fuzzy systems employ type-1 fuzzy sets, which represent uncertainty by numbers in the range [0, 1]. When an entity is uncertain, like a measurement, it is difficult to determine its exact value, and of course type-1 fuzzy sets make more sense than traditional sets [69]. However, it is not reasonable to use an accurate membership function for something uncertain, so in this case what we need is another type of fuzzy sets, those which are able to handle these uncertainties, the so called type-2 fuzzy sets [13]. The amount of uncertainty in a system can be reduced by using type-2 fuzzy logic because this logic offers better capabilities to handle linguistic uncertainties by modeling vagueness and unreliability of information [61,68].

Type-2 fuzzy models have emerged as an interesting generalization of fuzzy models based upon type-1 fuzzy sets [13,30]. There have been a number of claims put forward as to the relevance of type-2 fuzzy sets being regarded as generic building constructs of fuzzy models [26,59,64]. Likewise, there is a record of some experimental evidence showing some improvements in terms of accuracy of fuzzy models of type-2 over their type-1 counterparts [20,27,44]. Unfortunately, no systematic and comprehensive design framework has been provided and while improvements over type-1 fuzzy models were evidenced, it is not clear whether this effect could always be expected. Furthermore, it is not demonstrated whether the improvement is substantial enough and fully legitimized given the substantial optimization overhead associated with the design of this category of models. There have been a lot of applications of type-2 in intelligent control [9,14,30,31,50], pattern recognition [54], intelligent manufacturing [27,52,71], and others [2,17–19]. Similarly, optimization methods have also been applied in the design of optimal type-1 fuzzy systems in diverse areas of application [1,3–5,23,32,33,38,58]. However, no general design strategy for finding the optimal type-2 fuzzy model has been proposed, and for this reason bio-inspired algorithms have been used to try in find these optimal type-2 models.

In general, the methods for designing a type-2 fuzzy model based on experimental data can be classified into two categories. The first category of methods assumes that an optimal (in some sense) type-1 fuzzy model has already been designed and afterwards a type-2 fuzzy model is constructed through some sound augmentation of the existing model. The second class of design methods is concerned with the construction of the type-2 fuzzy model directly from experimental data. In both cases, an optimization method can help in obtaining the optimal type-2 fuzzy model for the particular application.

Recently, bio-inspired methods have emerged as powerful optimization algorithms for solving complex problems. In the case of designing type-2 fuzzy controllers for particular applications, the

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

1568-4946/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2011.12.010

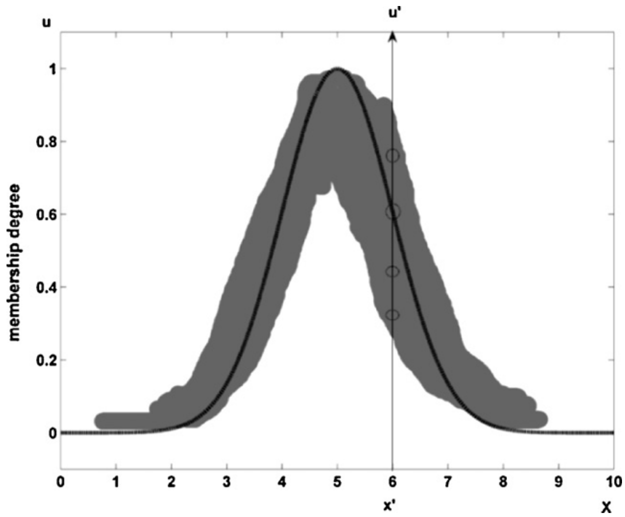


**Fig. 1.** Type-2 membership function as a blurred type-1 membership function.

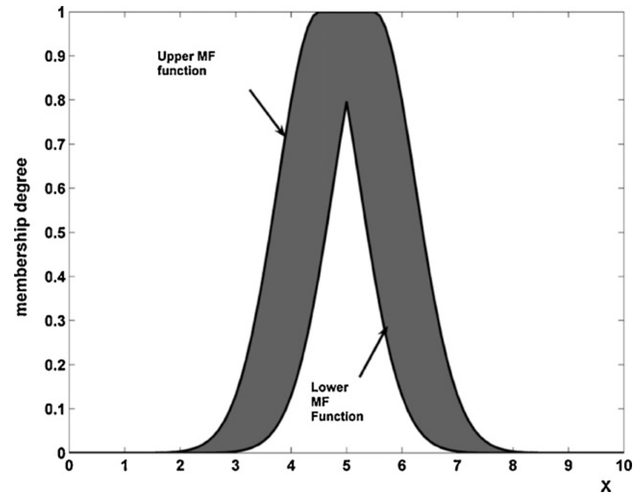


**Fig. 2.** Interval type-2 membership function.

use of bio-inspired optimization methods have helped in the complex task of finding the appropriate parameter values and structure of the fuzzy systems. In this review, we 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. We also mention some hybrid approaches and other optimization methods that have been applied in problem of designing optimal type-2 fuzzy controllers in different domains of application.

## 2. Type-2 fuzzy logic systems

In this section, a brief overview of type-2 fuzzy systems is presented. This overview is intended to provide the basic concepts needed to understand the methods and algorithms presented later in the paper [10,13].

If for a type-1 membership function, we blur it to the left and to the right, as illustrated in Fig. 1, then a type-2 membership function is produced. In this case, for a specific value $x'$, the membership function ($u'$), takes on different values, which are not all weighted the same, so we can assign membership grades to all of those points.

By doing this for all $x \in X$, we form a three-dimensional membership function – a type-2 membership function – that characterizes a type-2 fuzzy set [13]. A type-2 fuzzy set $\tilde{A}$, is characterized by the membership function:

$$\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u)) | \forall x \in X, \forall u \in J_x \subseteq [0, 1]\} \quad (1)$$

in which $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$. In fact $J_x \subseteq [0, 1]$ represents the primary membership of $x$, and $\mu_{\tilde{A}}(x, u)$ is a type-1 fuzzy set known as the secondary set. Hence, a type-2 membership grade can be any subset in [0, 1], the primary membership, and corresponding to each primary membership, there is a secondary membership (which can also be in [0, 1]) that defines the possibilities for the primary membership. Uncertainty is represented by a region, which is called the footprint of uncertainty (FOU). When $\mu_{\tilde{A}}(x, u) = 1, \forall u \in J_x \subseteq [0, 1]$ we have an interval type-2 membership function, as shown in Fig. 2. The uniform shading for the FOU represents the entire interval type-2 fuzzy set and it can be described in terms of an upper membership function $\bar{\mu}_{\tilde{A}}(x)$ and a lower membership function $\underline{\mu}_{\tilde{A}}(x)$.

An FLS described using at least one type-2 fuzzy set is called a type-2 FLS. Type-1 FLSs are unable to directly handle rule uncertainties, because they use type-1 fuzzy sets that are certain (viz., fully described by single numeric values). On the other hand, type-2 FLSs, are useful in circumstances where it is difficult to determine an exact numeric membership function, and there are measurement uncertainties.

A type-2 FLS is characterized by IF-THEN rules, where their antecedent or consequent sets are now of type-2. Type-2 FLSs, can be used when the circumstances are too uncertain to determine exact membership grades such as when the training data is affected by noise. Similarly, to the type-1 FLS, a type-2 FLS includes a fuzzifier, a rule base, fuzzy inference engine, and an output processor, as we can see in Fig. 3 (in this case, a fuzzy system with two inputs and one output is used as illustration). The output processor includes type-reducer and defuzzifier; it generates a type-1 fuzzy set output (from the type-reducer) or a number (from the defuzzifier) [13]. Now we explain each of the blocks shown in Fig. 3.

### 2.1. Fuzzifier

The fuzzifier maps a numeric vector $\mathbf{x} = (x_1, \ldots, x_p)^T \in X_1 \times X_2 \times \cdots \times X_p \equiv \mathbf{X}$ into a type-2 fuzzy set $\tilde{A}_x$ in $\mathbf{X}$ [13], an interval type-2 fuzzy set in this case. We use type-2 singleton fuzzifier, in a singleton fuzzification, the input fuzzy set has only a single point on nonzero membership. $\tilde{A}_x$ is a type-2 fuzzy singleton if $\mu_{\tilde{A}_x}(x) = 1/1$ for $\mathbf{x} = \mathbf{x}'$ and $\mu_{\tilde{A}_x}(x) = 1/0$ for all other $\mathbf{x} \neq \mathbf{x}'$.

### 2.2. Rules

The structure of rules in a type-1 FLS and a type-2 FLS is the same, but in the latter the antecedents and the consequents is represented by type-2 fuzzy sets. So for a type-2 FLS with $p$ inputs $x_1 \in X_1, \ldots, x_p \in X_p$ and one output $y \in Y$, Multiple Input Single Output (MISO), if we assume there are $M$ rules, the $l$th rule in the type-2 FLS can be written down as follows:

$$R^l: \quad \text{IF } x_1 \text{ is } \tilde{F}_1^l \text{ and} \cdots \text{and } x_p \text{ is } \tilde{F}_p^l, \quad \text{THEN } y \text{ is } \tilde{G}^l \quad l = 1, \ldots, M \quad (2)$$

### 2.3. Inference

In the type-2 FLS, the inference engine combines rules and gives a mapping from input type-2 fuzzy sets to output type-2 fuzzy sets. It is necessary to compute the join $\sqcup$, (unions) and the meet $\Pi$ (intersections), as well as the extended sup-star compositions (sup

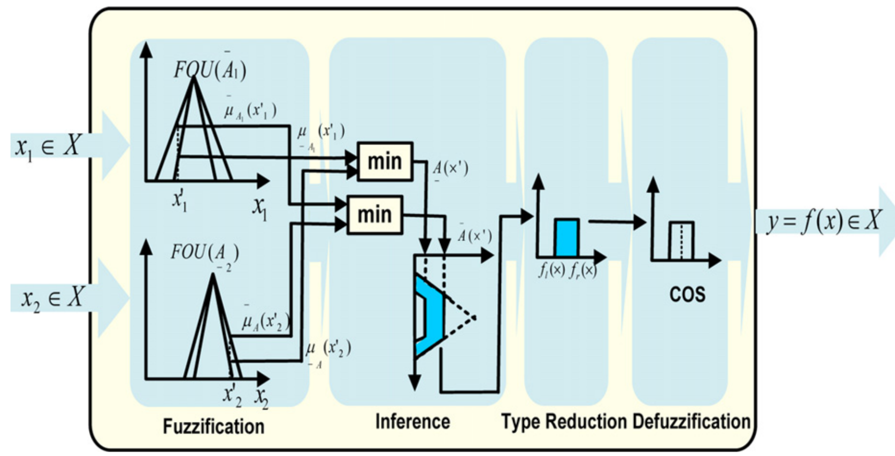


**Fig. 3.** Structure of a type-2 fuzzy logic system.

star compositions) of type-2 relations. If $\tilde{F}_1^l \times \cdots \times \tilde{F}_p^l = \tilde{A}^l$, (2) can be re-written as follows:

$$R^l: \quad \tilde{F}_1^l \times \cdots \times \tilde{F}_p^l \to \tilde{G}^l = \tilde{A}^l \to \tilde{G}^l \quad l = 1, \ldots, M \tag{3}$$

$R^l$ is described by the membership function $\mu_{R^l}(\mathbf{x}, y) = \mu_{R^l}(x_1, \ldots, x_p, y)$, where

$$\mu_{R^l}(\mathbf{x}, y) = \mu_{\tilde{A}^l \to \tilde{G}^l}(\mathbf{x}, y) \tag{4}$$

can be written as [14]:

$$\begin{aligned}\mu_{R^l}(\mathbf{x}, y) &= \mu_{\tilde{A}^l \to \tilde{G}^l}(\mathbf{x}, y) = \mu_{\tilde{F}_1^l}(x_1) \Pi \cdots \Pi \mu_{\tilde{F}_p^l}(x_p) \Pi \mu_{\tilde{G}^l}(y) \\ &= [\Pi_{i=1}^p \mu_{\tilde{F}_i^l}(x_i)] \Pi \mu_{\tilde{G}^l}(y)\end{aligned} \tag{5}$$

In general, the $p$-dimensional input to $R^l$ is given by the type-2 fuzzy set $\tilde{A}_x$ whose membership function becomes

$$\mu_{\tilde{A}_x}(\mathbf{x}) = \mu_{\tilde{x}_1}(x_1) \Pi \ldots \Pi \mu_{\tilde{x}p}(x_p) = \Pi_{i=1}^p \mu_{\tilde{x}i}(x_i) \tag{6}$$

where $\tilde{X}_i (i = 1, \ldots, p)$ are the labels of the fuzzy sets describing the inputs. Each rule $R^l$ determines a type-2 fuzzy set $\tilde{B}^l = \tilde{A}_x \circ R^l$ such that: $\mu_{\tilde{B}^l}(y) = \mu_{\tilde{A}_x \circ R^l} = \sqcup_{x \in \mathbf{X}} [\mu_{\tilde{A}_x}(\mathbf{x}) \sqcap \mu_{R^l}(\mathbf{x}, y)] \quad y \in Y \quad l = 1, \ldots, M$

This dependency is the input/output relation shown in Fig. 3, which holds between the type-2 fuzzy set that activates a certain rule in the inference engine and the type-2 fuzzy set at the output of that engine.

In the FLS, we used interval type-2 fuzzy sets and intersection under product t-norm, so the result of the input and antecedent operations, which are contained in the firing set $\Pi_{i=1}^p \mu_{\tilde{F}_i}(x_i' \equiv F^l(\mathbf{x}'))$, is an interval type-1 set,

$$F^l(\mathbf{x}') = \left[\underline{f}^l(\mathbf{x}'), \bar{f}^l(\mathbf{x}')\right] \equiv \left[\underline{f}^l, \bar{f}^l\right] \tag{8}$$

where

$$\underline{f}^l(\mathbf{x}') = \underline{\mu}_{\tilde{F}_p^l}(x_1') * \cdots * \underline{\mu}_{\tilde{F}_p^l}(x_p') \tag{9}$$

and

$$\bar{f}^l(\mathbf{x}') = \bar{\mu}_{\tilde{F}_1^l}(x_1') * \cdots * \bar{\mu}_{\tilde{F}_p^l}(x_p') \tag{10}$$

where * stands for the product operation.

### 2.4. Type reducer

The type-reducer generates a type-1 fuzzy set output, which is then converted in a numeric output through running the defuzzifier. This type-1 fuzzy set is also an interval set, for the case of our FLS we used center of sets (cos) type reduction, $Y_{\cos}$, which is expressed as

$$\begin{aligned}Y_{\cos}(\mathbf{x}) = [y_l, y_r] = \int_{y^1 \in [y_l^1, y_r^1]} \cdots \int_{y^M \in [y_l^M, y_r^M]} \int_{f^1 \in [\underline{f}^1, \bar{f}^1]} \cdots \\ \times \int_{f^M \in [\underline{f}^M, \bar{f}^M]} \frac{1}{\sum_{i=1}^M f^i y^i / \sum_{i=1}^M f^i}\end{aligned} \tag{11}$$

This interval set is determined by its two end points, $y_l$ and $y_r$, which corresponds to the centroid of the type-2 interval consequent set $\tilde{G}^i$,

$$C_{\tilde{G}^i} = \int_{\theta_1 \in J_{y1}} \cdots \int_{\theta_N \in J_{yN}} \frac{1}{\sum_{i=1}^N y_i \theta_i / \sum_{i=1}^N \theta_i} = [y_l^i, y_r^i] \tag{12}$$

before the computation of $Y_{\cos}(\mathbf{x})$, we must evaluate Eq. (12), and its two end points, $y_l$ and $y_r$. If the values of $f_i$ and $y_i$ that are associated with $y_l$ are denoted $f_l^i$ and $y_l^i$, respectively, and the values of $f_i$ and $y_i$ that are associated with $y_r$ are denoted $f_r^i$ and $y_r^i$, respectively, from Eq. (13), we have

$$y_l = \frac{\sum_{i=1}^M f_l^i y_l^i}{\sum_{i=1}^M f_l^i} \tag{13}$$

$$y_r = \frac{\sum_{i=1}^M f_r^i y_r^i}{\sum_{i=1}^M f_r^i} \tag{14}$$

The values of $y_l$ and $y_r$ define the output interval of the type-2 fuzzy system, which can be used to verify if training or testing data are contained in the output of the fuzzy system.

### 2.5. Defuzzifier

From the type-reducer, we obtain an interval set $Y_{\cos}$, to defuzzify it we use the average of $y_l$ and $y_r$, so the defuzzified output of an interval singleton type-2 FLS is

$$y(\mathbf{x}) = \frac{y_l + y_r}{2} \tag{15}$$

## 3. Bio-inspired optimization methods

In this section we present a brief overview of the basic concepts from bio-inspired optimization methods needed for this work.

**Table 1**
Basic nomenclature of PSO.

| Variable | Definition |
|---|---|
| $x_z^i$ | Particle position |
| $V_z^i$ | Particle velocity |
| $w_{ij}$ | Inertia weight |
| $p_z^i$ | Best "remembered" individual particle position |
| $p_z^g$ | Best "remembered" swarm position |
| $c_1, c_2$ | Cognitive and social parameters |
| $r_1, r_2$ | Random numbers between 0 and 1 |

### 3.1. Particle swarm optimization

Particle swarm optimization is a population based stochastic optimization technique inspired by social behavior of bird flocking or fish schooling [28]. PSO shares many similarities with evolutionary computation techniques such as the GA [62].

The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike the GA, the PSO has no evolution operators such as crossover and mutation. In the PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles [47]. Each particle keeps track of its coordinates in the problem space, which are associated with the best solution (fitness) it has achieved so far (the fitness value is also stored). This value is called *pbest*. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This location is called *lbest*. When a particle takes all the population as its topological neighbors, the best value is a global best and is called *gbest* [48].

The particle swarm optimization concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its *pbest* and *lbest* locations (local version of PSO). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward *pbest* and *lbest* locations [48]. In the past several years, PSO has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way when compared with other methods [46,47]. Another reason that PSO is attractive is that there are few parameters to adjust. One version, with slight variations, works well in a wide variety of applications. Particle swarm optimization has been considered for approaches that can be used across a wide range of applications, as well as for specific applications focused on a specific requirement [8,56].

The basic algorithm of PSO has the nomenclature defined in Table 1.

The equation to calculate the velocity is:

$$v_{z+1}^i = w_{ij} v_z^i + c_1 r_1 (p_z^i - x_z^i) + c_2 r_2 (p_z^g - x_z^i) \tag{16}$$

and the position of the individual particles is updated as follows:

$$x_{z+1}^i = x_z^i + v_{z=1}^i \tag{17}$$

The basic PSO algorithm is defined as shown in Table 2.

**Table 2**
Basic steps of the PSO algorithm.

Pseudocode of the PSO algorithm
1. Initialize
   a) Set constants $Z_{\max}$, $c_1$, $c_2$
   b) Randomly initialize particle position $x_0^i \in D$ in $R^n$ for $i = 1, \ldots, p$
   c) Randomly initialize particle velocities $0 \le vi0 \le v\max 0$ for $i = 1, \ldots, p$
   d) Set $Z = 1$
2. Optimize
   a) Evaluate function value *fik* using design space coordinates *xik*
   b) If $fiz \le fibest$ then $fibest = fiz$, $piz = xiz$
   c) If $fiz \le fgbest$ then $fgbest = fiz$, $pgz = xiz$
   d) If stopping condition is satisfied then go to 3
   e) Update all particle velocities *viz* for $i = 1, \ldots, p$
   f) Update al particle positions *xiz* for $i = 1, \ldots, p$
   g) Increment *z*
   h) Goto 2(a)
3. Terminate

### 3.2. Genetic algorithms

Genetic algorithms (GAs) are adaptive heuristic search algorithms based on the evolutionary ideas of natural selection and genetic processes [25]. The basic principles of GAs were first proposed by John Holland in 1975, inspired by the mechanism of natural selection, where stronger individuals are likely the winners in a competing environment [9,25]. GA assumes that the potential solution of any problem is an individual and can be represented by a set of parameters. These parameters are regarded as the genes of a chromosome and can be structured by a string of values in binary form. A positive value, generally known as a fitness value, is used to reflect the degree of "goodness" of the chromosome for the problem, which would be highly related with its objective value.

The pseudocode of a GA is defined in Table 3.

The simple procedure just described above is the basis for most applications of GAs found in the literature.

**Table 3**
Basic steps of the GA algorithm.

Pseudocode of the GA
1. Start with a randomly generated population of *n* chromosomes (candidate solutions to a problem).
2. Calculate the fitness of each chromosome in the population.
3. Repeat the following steps until *n* offspring have been created:
   a. Select a pair of parent chromosomes from the current population, the probability of selection being an increasing function of fitness. Selection is done with replacement, meaning that the same chromosome can be selected more than once to become a parent.
   b. With probability (crossover rate), perform crossover to the pair at a randomly chosen point to a form two offspring.
   c. Mutate the two offspring at each locus with probability (mutation rate), and place the resulting chromosomes in the new population.
4. Replace the current population with the new population.
5. Go to step 2.

### 3.3. Ant colony optimization

Ant colony optimization (ACO) is a probabilistic technique that can be used for solving problems that can be reduced to finding good paths along graphs. This method is inspired on the behavior presented by ants in finding paths from the nest or colony to the food source [28].

The S-ACO is an algorithmic implementation that adapts the behavior of real ants to solutions of minimum cost path problems on graphs [36]. A number of artificial ants build solutions for a certain optimization problem and exchange information about the quality of these solutions making allusion to the communication system of real ants [15].

Let us define the graph $G = (V, E)$, where $V$ is the set of nodes and $E$ is the matrix of the links between nodes. $G$ has $n_G = |V|$ nodes. Let us define $L^K$ as the number of hops in the path built by the ant $k$ from the origin node to the destiny node. Therefore, it is necessary to find:

$$Q = \{q_a, \ldots, q_f | q_1 \in C\} \tag{18}$$

where $Q$ is the set of nodes representing a continuous path with no obstacles; $q_a, \ldots, q_f$ are former nodes of the path and $C$ is the set of possible configurations of the free space. If $x^k(t)$ denotes a $Q$

Please cite this article in press as: O. Castillo, P. Melin, A review on the design and optimization of interval type-2 fuzzy controllers, Appl. Soft Comput. J. (2012), doi:10.1016/j.asoc.2011.12.010

**Table 4**
Basic steps of the ACO algorithm.

Pseudocode of ACO
1. Set a pheromone concentration $\tau_{ij}$ to each link $(i, j)$.
2. Place a number $k = 1, 2, \ldots, n_k$ in the nest.
3. Iteratively build a path to the food source (destiny node), using Eq. (19) for every ant.
   Remove cycles and compute each route weight $f(x^k(t))$. A cycle could be generated when there are no feasible candidates nodes, that is, for any $i$ and any $k$, $N_i^k = \varnothing$; then the predecessor of that node is included as a former node of the path.
4. Apply evaporation using Eq. (20).
5. Update of the pheromone concentration using Eq. (21).
6. Finally, finish the algorithm in any of the three different ways:
   When a maximum number of epochs has been reached.
   When it has found an acceptable solution, with $f(x_k(t)) < \varepsilon$.
   When all ants follow the same path.

solution in time $t$, $f(x^k(t))$ expresses the quality of the solution. The S-ACO algorithm is based on Eqs. (19)–(21):

$$p_{ij}^k(t) = \begin{cases} \dfrac{\tau_{ij}^k}{\sum_{j \in N_{ij}^k} \tau_{ij}^{\alpha}(t)} & \text{if } j \in N_i^k \\ 0 & \text{if } j \notin N_i^k \end{cases} \tag{19}$$

$$\tau_{ij}(t) \bullet (1-\rho)\tau_{ij}(t) \tag{20}$$

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \tau_{ij}(t) \tag{21}$$

Eq. (19) represents the probability for an ant $k$ located on a node $i$ selects the next node denoted by $j$, where, $N_i^k$ is the set of feasible nodes (in a neighborhood) connected to node $i$ with respect to ant $k$, $\tau_{ij}$ is the total pheromone concentration of link $ij$, and $\alpha$ is a positive constant used as a gain for the pheromone influence.

Eq. (20) represents the evaporation pheromone update, where $\rho \in [0, 1]$ is the evaporation rate value of the pheromone trail. The evaporation is added to the algorithm in order to force the exploration of the ants, and avoid premature convergence to sub-optimal solutions [37]. For $\rho = 1$ the search becomes completely random [48]. Eq. (21), represents the concentration pheromone update, where $\Delta\tau_{ij}^k$ is the amount of pheromone that an ant $k$ deposits in a link $ij$ in a time $t$.

The general steps of S-ACO are indicated in Table 4.

### 3.4. General remarks about optimization of type-2 fuzzy systems

The problem of designing type-2 fuzzy systems can be solved with any the above mentioned optimization methods. The main issue in any of these methods is deciding on the representation of the type-2 fuzzy system in the corresponding optimization paradigm. For example, in the case of GAs the type-2 fuzzy systems must be represented in the chromosomes. On the other hand, in PSO the fuzzy system is represented as a particle in the optimization process. In the ACO method, the fuzzy system can be represented as one of the paths that the ants can follow in a graph. Also, the evaluation of the fuzzy system must be represented as an objective function in any of the methods. In this paper, we offer a comprehensive review of the most representative works in optimization of type-2 fuzzy controllers that have been done around the world.

## 4. GAs in optimization of type-2 fuzzy controllers

There have been many works reported in the literature optimizing type-2 fuzzy controllers using different kinds of genetic algorithms. Most of these works have had relative success according to the different areas of application. In this section, we offer a representative review of these types of works to illustrate the advantages of using a bio-inspired optimization technique for automating the design process of type-2 fuzzy controllers.

In a paper by N. Cazarez et al. [21] a genetic-type-2 fuzzy logic controller was proposed to achieve the output regulation of a servomechanism with backlash. The problem of designing the type-2 fuzzy controller was solved by optimizing the parameters of the fuzzy system with a genetic algorithm to obtain the closed-loop system in which the load of the driver is regulated to a desired position. The provided servomotor position is the only measurement available for feedback. Simulations results illustrate the effectiveness of the optimized closed-loop system.

In the work of C. Wagner and H. Hagras [62,63], a genetic algorithm for evolving type-2 fuzzy logic controllers for real world autonomous robots was presented. The type-2 fuzzy logic controller (FLC) has started to emerge as a promising control mechanism for autonomous mobile robots navigating in real world environments. This is because such robots need control mechanisms such as type-2 FLCs which can handle the large amounts of uncertainties present in real world environments. However, manually designing and tuning the type-2 membership functions (MFs) for an interval type-2 FLC to give a good response is a difficult task. This work describes a genetic algorithm to evolve the type-2 MFs of interval type-2 FLCs for mobile robots that will navigate in real world environments. The evolved type-2 FLCs dealt with the uncertainties present in the real world to give a very good performance that has outperformed their type-1 counterparts as well as the manually designed type-2 FLCs.

In the work by D. Wu and W. Tan [67] genetic learning and performance evaluation of interval type-2 fuzzy logic controllers was presented. This paper focuses on advancing the understanding of interval type-2 fuzzy logic controllers (FLCs). First, a type-2 FLC was evolved using genetic algorithms. The type-2 FLC was then compared with another three GA evolved type-1 FLCs that have different design parameters. The objective was to examine the amount by which the extra degrees of freedom, provided by antecedent type-2 fuzzy sets, was able to improve the control performance. Experimental results show that better control can be achieved using a type-2 FLC with fewer fuzzy sets/rules so one benefit of type-2 FLC was a lower trade-off between modeling accuracy and interpretability.

The work by D. Wu and W. Tan [66] focuses on evolving type-2 fuzzy logic controllers genetically and examining whether they are better able to handle modeling uncertainties. The study was conducted by utilizing a type-2 FLC, evolved by a genetic algorithm, to control a liquid-level process. A two stage strategy is employed to design the type-2 FLC. First, the parameters of a type-1 FLC are optimized using the GA. Next, the footprint of uncertainty was evolved by blurring the fuzzy input set. Experimental results show that the type-2 FLC copes well with the complexity of the plant, and can handle the modeling uncertainty better than its type-1 counterpart.

In the work by C. H. Wang et al. [65] a type-2 fuzzy logic system cascaded with neural network, type-2 fuzzy neural network (T2FNN), was presented to handle uncertainty with dynamical optimal learning. A T2FNN consists of a type-2 fuzzy linguistic process as the antecedent part, and the two-layer interval neural network as the consequent part. A general T2FNN is computational-intensive due to the complexity of type-2 to type-1 reduction. Therefore, the interval T2FNN is adopted in this work to simplify the computational process. The dynamical optimal training algorithm for the two-layer consequent part of interval T2FNN was first developed. To achieve better total performance, a genetic algorithm was designed to search optimal spread rate for uncertain means and optimal learning for the antecedent part. Several examples are fully illustrated. Excellent results are obtained for the truck backing-up control and the identification of nonlinear system, which yield more improved performance than those using type-1 FNN.

In the work by L. Cervantes and O. Castillo [22] a genetic design of a fuzzy system for the longitudinal control of an F-14 airplane was presented. The longitudinal control is carried out only by controlling the elevators of the airplane. To carry out such monitoring it is necessary to use the stick, the rate of elevation and the angle of attack. Simulation results of the longitudinal control are obtained using a plant in Simulink and those results were compared against the PID controller. Genetic algorithms were used to optimize parameters of type-2 and type-1 fuzzy systems to find the best fuzzy controller under noisy conditions. The type-2 fuzzy controller outperforms the type-1 when the level of noise is sufficiently high.

In the work of W.-D. Kim et al. [39], a design methodology of an optimized type-2 fuzzy cascade controller with the aid of hierarchical fair competition-based genetic algorithm (HFCGA) for ball and beam system was proposed. The type-2 fuzzy cascade controller scheme consists of the outer controller and the inner controller as two cascaded fuzzy controllers. In type-2 fuzzy logic controller (FLC) as the expanded type of type-1 fuzzy logic controller (FLC), we can effectively improve the control characteristic by using the footprint of uncertainty (FOU) of membership function. The control parameters (scaling factors) of each fuzzy controller are obtained using HFCGA which is a kind of parallel genetic algorithms (PGAs). HFCGA helps alleviate the premature convergence being generated in conventional genetic algorithms (GAs). The controller characteristic parameters of optimized type-2 fuzzy cascade controller applied ball & beam system such as maximum overshoot, delay time, rise time, settling time and steady-state error were estimated. For a detailed comparative analysis from the viewpoint of the performance results and the design methodology, the proposed method for the ball & beam system which is realized by the fuzzy cascade controller based on HFCGA, was presented in comparison with the conventional PD cascade controller based on serial genetic algorithms.

In the work of R. Martinez et al. [45], a tracking controller for the dynamic model of a unicycle mobile robot by integrating a kinematic and a torque controller based on type-2 fuzzy logic theory and genetic algorithms was proposed. Genetic optimization enables finding the optimal parameters of the type-2 fuzzy controller for the mobile robot. Computer simulations are presented confirming the performance of the tracking controller and its application to different navigation problems.

In this work by C. Li et al. [42], a single-input-rule-modules (SIRMs) based type-2 fuzzy logic control scheme was proposed for a particular nonlinear multivariable system, which is in this case the translational oscillation with a rotational proof-mass actuator (TORA). Genetic algorithms (GAs) are adopted to determine the parameters and to improve the performance of the SIRMs based type-2 fuzzy logic controller (SIRM-T2FLC). At last, simulations and comparisons are given to demonstrate the effectiveness, robustness and superiority of the proposed controller under three circumstances: normal case, the disturbance existing case, and the parameter varying case. From the design process and comparisons, it was evident that: (1) this SIRMs based type-2 fuzzy control scheme can alleviate the difficulty to design conventional type-2 fuzzy logic controllers (T2FLCs) for this multivariable TORA system, (2) the SIRM-T2FLC is much easier to design and understand compared with conventional nonlinear control strategies for the TORA system, (3) better performance can be achieved.

In Table 5 a summary of the previously presented contributions, were GAs has been applied to optimize type-2 fuzzy controllers, is presented. The comparison shown in Table 5 is based on the following criteria: author names, year of publication, reference number, if a comparison with type-1 fuzzy logic is provided, if a comparison with other optimization methods is presented, and why type-2 fuzzy logic was used by the authors.

## 5. PSO in optimization of type-2 fuzzy controllers

There have been several works reported in the literature optimizing type-2 fuzzy controllers using different kinds of PSO algorithms. Most of these works have had relative success according to the different areas of application. In this section, we offer a representative review of these types of works to illustrate the advantages of using the PSO optimization technique for automating the design process of type-2 fuzzy systems.

In the work of J. Cao et al. [8], the PSO algorithm was used to derive an adaptive fuzzy logic controller (AFC) based on interval fuzzy membership functions for vehicle non-linear active suspension systems. The interval membership functions were utilized in the AFC design to deal with not only non-linearity and uncertainty caused from irregular road inputs and immeasurable disturbance, but also the potential uncertainty of expert's knowledge and experience. The adaptive strategy was designed to self-tune the active force between the lower bounds and upper bounds of interval fuzzy outputs. A case study based on a quarter active suspension model demonstrated that the proposed adaptive fuzzy controller significantly outperforms conventional fuzzy controllers of an active suspension and a passive suspension.

In the work by R. Martinez et al. [47], bio-inspired optimization methods were applied to design type-2 fuzzy logic controllers to minimize the steady state error of linear plants. The optimal type-2 fuzzy controllers obtained by genetic algorithms and PSO were compared using benchmark plants. The bio-inspired methods were used to find the parameters of the membership functions of the type-2 fuzzy system to obtain the optimal controller. Simulation results were presented to show the feasibility of the proposed approach.

In this work by S.-K. Oh et al. [56], the design methodology of an optimized fuzzy controller with the aid of PSO for the ball and beam system was introduced. This system exhibits a number of interesting and challenging properties when being considered from the control perspective. The ball and beam system determines the position of ball through the control of a servo motor. The displacement change of the position of ball leads to the change of the angle of the beam which determines the position angle of a servo motor. The fixed membership function design of type-1 based fuzzy logic controller (FLC) leads to the difficulty of rule-based control design when representing linguistic nature of knowledge. In type-2 FLC as the expanded type of type-1 FL, we can effectively improve the control characteristic by using the footprint of uncertainty (FOU) of the membership functions. Type-2 FLC exhibits some robustness when compared with type-1 FLC.

In this work of Z. Bingül and O. Karahan [7], a 2 DOF planar robot was controlled by Fuzzy Logic Controller tuned with a particle swarm optimization. For a given trajectory, the parameters of Mamdani-type-Fuzzy Logic Controller (the centers and the widths of the Gaussian membership functions in inputs and output) were optimized by the particle swarm optimization with three different cost functions. In order to compare the optimized Fuzzy Logic Controller with different controller, the PID controller was also tuned with particle swarm optimization. In order to test the robustness of the tuned controllers, the model parameters and the given trajectory were changed and the white noise was added to the system. The simulation results show that fuzzy logic controller tuned by particle swarm optimization is better and more robust than the PID tuned by particle swarm optimization for robot trajectory control.

**Table 5**
GAs for the optimization of type-2 fuzzy controllers.

| Author (s) (pub. year) | Ref. no. | Comparison with type-1 | Comparison with other optimization | Why type-2 is required for the problem? |
|---|---|---|---|---|
| Cazarez et al. (2008) | [21] | Yes | No | Uncertainty in control |
| Wagner and Hagras (2007) | [62][63] | Yes | No | Uncertainty in control |
| Wu and Tan (2006) | [67] | Yes | No | Testing type-2 fuzzy control |
| Wu and Tan (2004) | [66] | Yes | No | Uncertainty in control |
| Wang et al. (2004) | [65] | Yes | Yes | Uncertainty in control |
| Cervantes and Castillo (2010) | [22] | Yes | No | Uncertainty in control |
| Kim et al. (2010) | [39] | Yes | Yes | Testing type-2 fuzzy control |
| Martinez et al. (2009) | [45] | Yes | Yes | Uncertainty in robot control |
| Li et al. (2009) | [42] | Yes | Yes | Uncertainty in control |

This work by O. Linda and M. Manic [43], presents a comparative analysis of interval T2 (IT2) and T1 FLCs in the context of learning behaviors for mobile robotics. First, a T1 FLC is optimized using the PSO algorithm to mimic a wall-following behavior performed by an operator. Next, an IT2 FLC is constructed by symmetrically blurring the fuzzy sets of the original T1 FLC. The performance of the fuzzy controllers is compared using a wall-following sonar-equipped mobile robot in both noise-free and noisy environments. It is experimentally demonstrated that the IT2 FLC can cope better with dynamic uncertainties in the sensory inputs due to the softening and smoothing of the output control surface by the IT2 fuzzy sets.

In the work by R. Martinez et al. [46], the optimization of type-2 fuzzy logic controllers using PSO was presented. The PSO method was applied to find the parameters of the membership functions of a type-2 fuzzy logic controller in order to minimize the state error for linear systems. PSO was used to find the optimal type-2 fuzzy controller to achieve regulation of the output and stability of the closed-loop system. For this purpose, the values of the cognitive, social and inertia variables in the PSO were changed. Simulation results, with the optimal type-2 fuzzy controller implemented in Simulink, show the potential applicability of the proposed approach.

In Table 6 a summary of the previously presented contributions, were PSO has been applied to optimize type-2 fuzzy controllers, is presented. The comparison shown in Table 6 is based on the following criteria: author names, publication year, reference number, if a comparison with type-1 fuzzy logic is provided, if a comparison with other optimization methods is presented, and why type-2 fuzzy logic was used by the authors. It can also be noted that the number of papers using PSO is lower than the ones using GAs, mentioned in the previous section.

## 6. ACO in optimization of type-2 fuzzy controllers

There have been several works reported in the literature optimizing type-2 fuzzy controllers using different kinds of ACO algorithms. Most of these works have had relative success according to the different areas of application. In this section, we offer a representative review of these types of works to illustrate the advantages of using the ACO optimization technique for automating the design process or parameters of type-2 fuzzy controllers.

In the work of C.F. Juang et al. [37], a reinforcement self-organizing interval type-2 fuzzy system with ant colony optimization (RSOIT2FS-ACO) method was proposed. The antecedent part in each fuzzy rule of the RSOIT2FS-ACO uses interval type-2 fuzzy sets in order to improve system robustness to noise. The consequent part of each fuzzy rule was designed using the ACO technique. The RSOIT2FS-ACO method was applied to a truck backing control. The proposed RSOIT2FS-ACO was compared with other reinforcement fuzzy systems to verify its efficiency and effectiveness. A comparison with type-1 fuzzy systems verifies the robustness of using type-2 fuzzy systems to noise.

In the work of R. Martinez-Marroquin et al. [48], the application of a simple ACO as an optimization method for the membership functions' parameters of a fuzzy logic controller was proposed. The application of ACO enables finding the optimal intelligent controller for an autonomous wheeled mobile robot. In the ACO implementation, each controller was represented as a trajectory on a graph. Simulation results show that ACO outperforms a GA in the optimization of type-2 fuzzy logic controllers for a particular autonomous mobile robot.

In the work of C.F. Juang and C.H. Hsu [35], a reinforcement ant optimized fuzzy controller (FC) design method, called RAOFC, was proposed. The method was applied it to wheeled-mobile-robot wall-following control under reinforcement learning environments. The inputs to the designed FC are range-finding sonar sensors, and the controller output is a robot steering angle. The consequent part of each fuzzy rule is designed using *Q*-value aided ant colony optimization (QACO). Simulations and experiments on mobile-robot wall-following control show the effectiveness and efficiency of the proposed RAOFC.

In the work of O. Castillo et al. [15], the application of ACO and PSO for the optimization of an interval type-2 fuzzy logic controller for an autonomous wheeled mobile robot was presented. The obtained simulation results were statistically compared with the obtained previous work results with GAs in order to determine the best optimization technique for this particular robotics problem. Both PSO and ACO were able to outperform GAs for this particular application. In comparing ACO and PSO, the best results were achieved with ACO.

In the work of C.F. Juang and C.H. Hsu [36], a new reinforcement-learning method using online rule generation and Q-value-aided ant colony optimization (ORGQACO) for fuzzy controller design was proposed. The fuzzy controller is based on an interval type-2 fuzzy system (IT2FS). The antecedent part in the designed IT2FS uses interval type-2 fuzzy sets to improve controller robustness to noise. The ORGQACO concurrently designs both the structure and parameters of an IT2FS. The ORGQACO design method was applied to the following three control problems: (1) truck-backing control; (2) magnetic-levitation control; and (3) chaotic-system control.

In Table 7 a summary of the previously presented contributions, were ACO has been applied to optimize type-2 fuzzy controllers, is presented. The comparison shown in Table 7 is based on the following criteria: author names, year of publication, reference number, if a comparison with type-1 fuzzy logic is provided, if a comparison with other optimization methods is presented, and why type-2 fuzzy logic was used by the authors. It can also be noted that the number of papers mentioning the use of ACO is lower than the ones using PSO or GAs.

## 7. Other methods for design and optimization of type-2 fuzzy controllers

In this section we describe some other works reported in the literature optimizing type-2 fuzzy systems using other of

**Table 6**
PSO for type-2 fuzzy controller optimization.

| Author (s) (pub. year) | Ref. no. | Comparison with type-1 | Comparison with other optimization | Why type-2 is required for the problem? |
|---|---|---|---|---|
| Cao et al. (2008) | [8] | Yes | No | Uncertainty in control |
| Martinez et al. (2010) | [47] | Yes | Yes | Uncertainty in control |
| Oh et al. (2011) | [56] | Yes | Yes | Testing type-2 fuzzy control |
| Bingül and Karahan (2011) | [7] | No | No | Presence of noise |
| Linda and Manic (2010) | [43] | Yes | Yes | Testing type-2 fuzzy control |
| Martinez et al. (2010) | [46] | Yes | Yes | Testing type-2 fuzzy control |

optimization or design methods. Most of these works have had relative success according to the different areas of application. In this section, we offer a representative review of these types of works to illustrate the advantages of using the corresponding method for automating the design process or parameters of type-2 fuzzy controllers.

In the work by S.M.A. Mohammadi et al. [55], an evolutionary tuning technique for type-2 fuzzy logic controller was presented. This work deals with the parameter optimization of the type-2 fuzzy membership functions using a new proposed reinforcement learning algorithm in a nonlinear system. The performance of the proposed method on initial error reduction and error convergence issues were also investigated by computer simulations.

In the work of O. Castillo et al. [16], a method for designing optimal interval type-2 fuzzy logic controllers using evolutionary algorithms was presented. An evolutionary algorithm is applied to find the optimal interval type-2 fuzzy system as mentioned above. The human evolutionary model is applied for optimizing the interval type-2 fuzzy controller for a particular non-linear plant and results are compared against an optimal type-1 fuzzy controller. A comparative study of simulation results of the type-2 and type-1 fuzzy controllers, under different noise levels, was also presented. Simulation results show that interval type-2 fuzzy controllers obtained with the evolutionary algorithm outperform type-1 fuzzy controllers.

In the work by K.J. Poornaselvan et al. [57], the main objective was to focus on an agent based approach to flight control in ground/runway. The idea was to provide an autonomous control on flight once the airplane comes to runway. In all airports there is a particular structure for the runway, like main runways, sub runways, different tracks. An interval type 2 fuzzy controller can be applied to the autonomous vehicle in order to handle uncertainty in a better way. Ant colony optimization technique can be used for an optimized path planning in traffic environment with more number of flights. A hybrid ant colony optimization was used to handle real time dynamic environment and path planning.

In the work by O. Castillo et al. [11], an evolutionary computing based approach for the optimization of type-2 fuzzy systems was presented. In particular, the application of hierarchical genetic algorithms for fuzzy system optimization in intelligent control was proposed. The problem of optimizing the number of rules and membership functions using an evolutionary approach was considered. The hierarchical genetic algorithm enables the optimization of the fuzzy system design for a particular application. The proposed approach was illustrated with the case of intelligent control in a medical application. Simulation results for this application show that an optimal set of rules and membership functions for the fuzzy system can be obtained with this approach.

In the work by L. Astudillo et al. [2], an optimization method based on the chemical reaction paradigms was proposed. The new optimization method was inspired on a nature based paradigm: the reaction methods existing on chemistry, and the way the elements combine with each other to form compounds, in other words, quantum chemistry. The proposed optimization method was tested with benchmark mathematical functions and also with the design of type-2 fuzzy controllers.

In the work by O. Castillo et al. [12], the use of hierarchical genetic algorithms for type-2 fuzzy system optimization in anesthesia control was proposed. In particular, the problem of optimizing the number of rules and membership functions using an evolutionary approach was proposed. The hierarchical genetic algorithm enables the optimization of the type-2 fuzzy system design for the particular application of anesthesia control. Simulation results for this application show that an optimal set of rules and membership functions for the type-2 fuzzy controller.

The work by M. Biglarbegian et al. [6], presents a novel design methodology of interval type-2 Takagi Sugeno Kang fuzzy logic controllers (IT2 TSK FLCs) for modular and reconfigurable robots (MRR) manipulators with uncertain dynamic parameters. A mathematical framework for the design of IT2 TSK FLCs for tracking purposes that can be effectively used in real-time applications was developed. To verify the effectiveness of the proposed controller, experiments on an MRR with two degrees of freedom were performed, which exhibits dynamic coupling behavior. Results show that the developed controller can outperform some well-known linear and nonlinear controllers for different configurations.

This work of G.O. Koca et al. [40], describes a new control scheme for the robust crank angular speed control of a four-bar mechanism driven by a DC motor. The proposed control method is based on type-2 fuzzy logic and sliding mode control (SMC) technique. Type-2 fuzzy control (FC) systems are characterized by type-2 membership functions which are useful in circumstances where it is difficult to determine an exact membership function. SMC is a robust control method against parameter variations and external disturbances. Type-2 fuzzy logic and SMC can be combined to use the advantages of both methods and thus to improve the effectiveness of the controllers. One of the most important advantages of the use of SMC with type-2 FC is to reduce the number of fuzzy rules and thus to obtain a simpler and more practical control algorithm to use in real applications.

**Table 7**
ACO optimization of type-2 fuzzy controllers.

| Author (s) (pub. year) | Ref. no. | Comparison with type-1 | Comparison with other optimization | Why type-2 is required for the problem? |
|---|---|---|---|---|
| Juang et al. (2009) | [37] | Yes | No | Uncertainty in control |
| Martinez-Marroquin et al. (2009) | [48] | Yes | Yes | Uncertainty in mobile robots |
| Juang and Hsu (2009) | [35] | No | No | Uncertainty in navigation |
| Castillo et al. (2011) | [15] | Yes | Yes | Modeling uncertainty in control |
| Juang and Hsu (2009) | [36] | Yes | No | Test different controllers |

This work of K.R. Sudha et al. [60], proposes a type-2 (T2) fuzzy approach for load frequency control of two-area interconnected reheat thermal power system with the consideration of generation rate constraint (GRC). The performance of the type-2 (T2) controller is compared with conventional controller and type-1 (T1) fuzzy controller with regard to generation rate constraint (GRC). The load frequency control (LFC) problem has been a major subject in electrical power system design/operation. LFC is becoming more significant recently with increasing size, changing structure and complexity in interconnected power systems. In practice LFC systems use simple proportional integral (PI) controllers. As the PI control parameters are usually tuned, based on classical approaches.

The work of C.-H. Lee et al. [41], proposes a new control scheme using type-2 fuzzy neural network (type-2 FNN) and adaptive filter for controlling nonlinear uncertain systems. This type-2 FNN model combines the advantages of type-2 fuzzy logic systems and neural networks. The type-2 FNN system has the ability of universal approximation, that is, identification of nonlinear dynamic systems. The proposed control scheme consists of a PD-type adaptive FNN controller and a pre-filter. The adaptive filter is used to provide better performance under transient response and to treat the problem of disturbance attenuation.

This work by M. Zaher and H. Hagras [70], describes a method to generate a type-2 fuzzy logic model entirely from data to provide a dynamic footprint of uncertainty for the generated fuzzy set. The fuzzy model was used to predict the wind speed experienced by a wind turbine without the use of sensors. This estimated wind speed is then passed for another fuzzy controller that changes the pitch angles of the wind turbine blades in order to track the maximum power available. Wind energy is becoming one of the most important and promising areas of renewable energy. During the past few years, wind energy generation underwent strong improvements in several fields including power electronics, mechanics, wind dynamics, etc.

In the work of M. Galluzzo and B. Cosenza [29], two adaptive type-2 fuzzy logic controllers with minimum number of rules are developed and compared by simulation for control of a bioreactor in which aerobic alcoholic fermentation for the growth of Saccharomyces cerevisiae takes place. The bioreactor model is characterized by nonlinearity and parameter uncertainty. The first adaptive fuzzy controller is a type-2 fuzzy-neuro-predictive controller (T2FNPC) that combines the capability of type-2 fuzzy logic to handle uncertainties, with the ability of predictive control to predict future plant performance making use of a neural network model of the nonlinear system. The second adaptive fuzzy controller is instead a self-tuning type-2 PI controller, where the output scaling factor is adjusted online by fuzzy rules according to the current trend of the controlled process.

This work by E.A. Jammeh et al. [34], proposes an interval type-2 FLC that achieves a superior delivered video quality compared with existing traditional controllers and a T1 FLC. To show the response in different network scenarios, tests demonstrate the response both in the presence of typical Internet cross-traffic as well as when other video streams occupy a bottleneck on an All-Internet protocol (IP) network. It was found that the proposed type-2 FLC, although it is specifically designed for Internet conditions, can also successfully react to the network conditions of an All-IP network. When the control inputs were subject to noise, the type-2 FLC resulted in an order of magnitude performance improvement in comparison with the T1 FLC.

In this work by H. Chaoui and W. Gueaieb [24], a type-2 fuzzy logic controller (FLC) is proposed in this article for robot manipulators with joint elasticity and structured and unstructured dynamical uncertainties. The proposed controller is based on a sliding mode control strategy. To enhance its real-time performance, simplified interval fuzzy sets are used. The efficiency of the control scheme is further enhanced by using computationally inexpensive input signals independently of the noisy torque and acceleration signals, and by adopting a trade off strategy between the manipulator's position and the actuators' internal stability. The controller is validated through a set of numerical experiments and by comparing it against its type-1 counterpart.

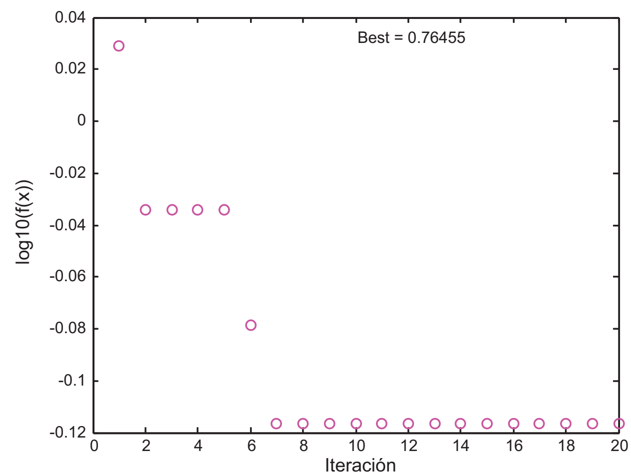


**Fig. 4.** Optimization behavior for the S-ACO on type-2 FLC optimization.

In this work by P. Melin and O. Castillo [51], adaptive model-based control of non-linear plants using type-2 fuzzy logic and neural networks was proposed. First, the general concept of adaptive model-based control is described. Second, the use of type-2 fuzzy logic for adaptive control is described. Third, a neuro-fuzzy approach is proposed to learn the parameters of the fuzzy system for control. A specific non-linear plant was used to simulate the hybrid approach for adaptive control. The non-linear plant that was considered is the "Pendubot", which is similar to the two-link robot arm. The results of the type-2 fuzzy logic approach for control were good, both in accuracy and efficiency.

In Table 8 a summary of the contributions were other optimization methods (different than GAs, PSO and ACO) have been applied to design type-2 fuzzy systems is presented. The comparison is based on the optimization method used, if a comparison with type-1 fuzzy logic is provided, and if a comparison with other optimization methods is presented.

## 8. Simulation results illustrating the optimization of type-2 fuzzy controllers

In this section we describe as an illustration the application of ACO for the optimization of the membership functions' parameters of a type-2 fuzzy logic controller in order to find the optimal intelligent controller for an autonomous wheeled mobile robot. The complete details of the robot, the fuzzy controller and simulation results can be found in [15].

Fig. 4 shows the optimization behavior of the ACO method. Fig. 5 shows the membership functions of the FLC obtained by the simple ACO algorithm. Fig. 6 shows both the desired trajectory and obtained trajectory for the robot.

A trajectory tracking controller was designed based on the dynamics and kinematics of the autonomous mobile robot through the application of ACO for the optimization of membership functions for the type-1 and type-2 fuzzy controllers with good results obtained after simulations. In summary, the three optimization methods are able to optimize the fuzzy controllers (to a certain

**Table 8**
Other methods of optimization for type-2 fuzzy controllers.

| Author (s) (pub. year) | Ref. no. | Comparison with type-1 | Comparison with other optimization | Optimization method |
|---|---|---|---|---|
| Mohammadi et al. (2010) | [55] | Yes | No | Evolutionary Algorithm |
| Castillo et al. (2011) | [16] | Yes | No | Human Evolutionary Model |
| Poornaselvan et al. (2008) | [57] | No | No | Hybrid Agent and ACO |
| Castillo et al. (2007) | [11] | Yes | No | Evolutionary Algorithm |
| Astudillo et al. (2010) | [2] | Yes | Yes | Chemical Optimization |
| Castillo et al. (2008) | [12] | Yes | No | Hierarchical Evolution |
| Biglarbegian et al. (2011) | [6] | Yes | No | Traditional Optimization |
| Koca et al. (2011) | [40] | No | No | Traditional Optimization |
| Sudha et al. (2011) | [60] | Yes | No | Traditional Optimization |
| Lee et al. (2005) | [41] | No | No | Neuro-Fuzzy Design |
| Zaher and Hagras (2010) | [70] | No | No | Traditional Optimization |
| Galluzzo and Cosenza (2010) | [29] | No | No | Neuro-Fuzzy Design |
| Jammeh et al. (2009) | [34] | Yes | No | Traditional Optimization |
| Chaoui and Gueaieb (2008) | [24] | Yes | No | Traditional Optimization |
| Melin and Castillo (2004) | [51] | Yes | No | Neuro-Fuzzy Design |

level) and the difference is in the achieved tracking errors, which are lower for ACO [15].

## 9. General overview of the area and future trends

Fig. 7 shows the total number of papers published per year describing the application of optimization methods for designing type-2 fuzzy controllers. From Fig. 7 it can be noted that the number of papers published have been increasing each year (in 2011 there is a decline because the information of this year is not complete at the moment of writing the paper). It is expected that this increasing trend will continue in the future because type-2 fuzzy systems have been used more frequently in the applications, and this will require designing more complex type-2 fuzzy systems, which in turn will need even better optimization techniques. It is also worth mentioning that at the moment most of the type-2 fuzzy controllers considered in the applications only use interval type-2 fuzzy sets, but when generalized type-2 fuzzy sets become more of a standard the design problem would require even more powerful optimization techniques.

Fig. 8 shows the distribution of the published papers in optimizing type-2 fuzzy controllers with the different techniques mentioned previously. From Fig. 8 it can be noted that the use of GAs have been decreasing, on the other hand the use of PSO, ACO and other methods have been increasing. Regarding the question of which method would be the most appropriate for optimizing type-2 fuzzy controllers, there is no easy answer. What we can be sure of is that the techniques mentioned in this paper and newer ones that may appear in the future, would certainly be tested in the optimization of type-2 fuzzy controllers because the problem of designing automatically these type of systems is complex enough to require their use.

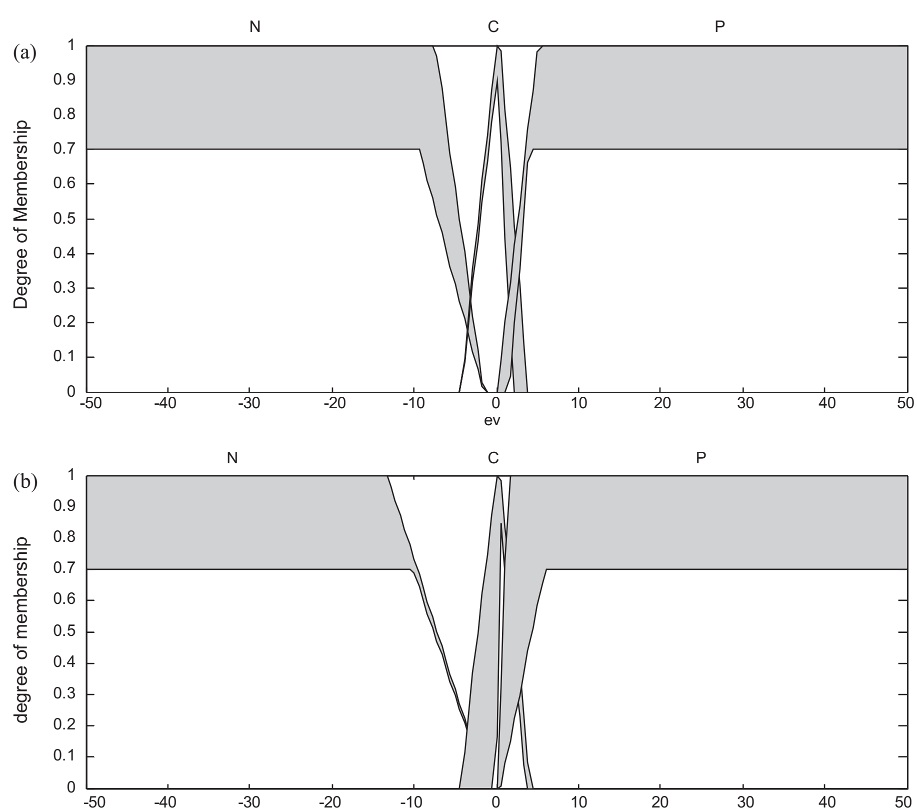


**Fig. 5.** Membership functions: (a) linear velocity error, (b) angular velocity error optimized by the ACO algorithm.

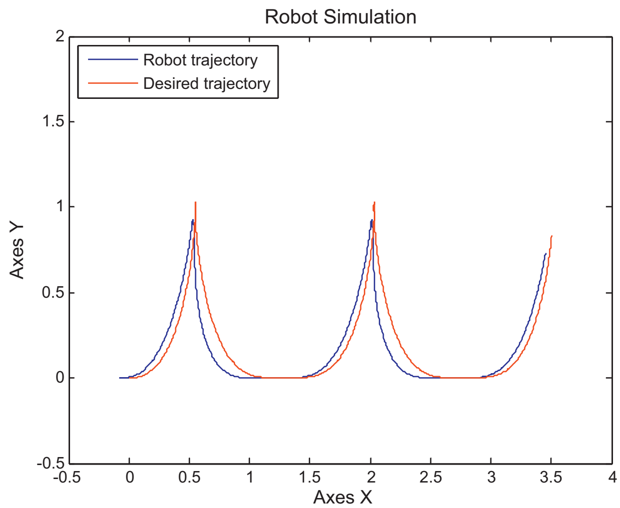


**Fig. 6.** Obtained trajectory with type-2 FLC optimization.

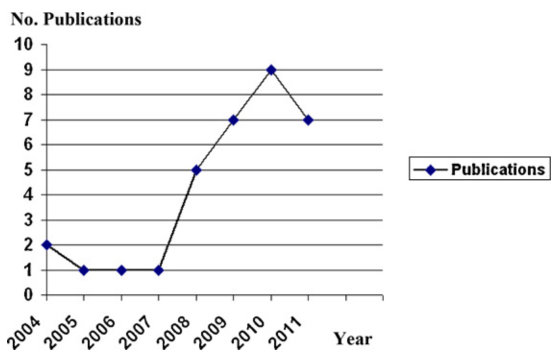


**Fig. 7.** Total publications per year for the 2004–2011 period of time.

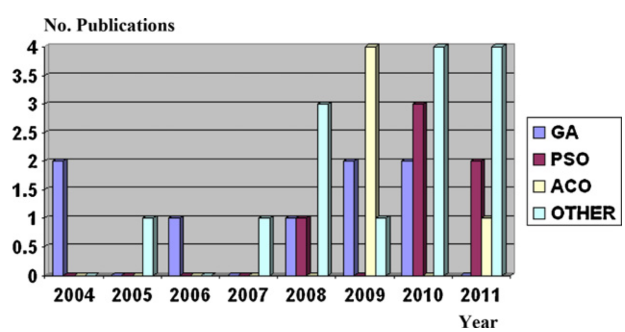


**Fig. 8.** Distribution of publications per area and year.

## 10. Conclusions

In the previous sections we have presented a representative account of the different optimization methods that have been applied in the optimal design of type-2 fuzzy systems. To the moment, genetic algorithms have been used more frequently to optimize type-2 fuzzy systems. However, more recently PSO and ACO have attracted more attention and have also been applied with some degree of success to the problem of optimal design of type-2 fuzzy systems. There have been also other optimization methods applied to the optimization of type-2 fuzzy controllers, like the chemical optimization paradigm. At this time, it would be very difficult to declare one of these optimization techniques as the best for optimizing type-2 fuzzy systems, as different techniques have had success for different applications of type-2 fuzzy logic. In any case, the need for bio-inspired optimization methods is justified due to the complexity of designing type-2 fuzzy systems.

There are other bio-inspired or nature-inspired techniques that at the moment have not been applied to the optimization of type-2 fuzzy systems that may be worth mentioning. For example, membrane computing, harmony computing, electromagnetism based computing, and other similar approaches have not been applied (to the moment) in the optimization of type-2 fuzzy systems. It is expected that these approaches and similar ones could be applied in the near future in the area of type-2 fuzzy system optimization. Of course, as new bio-inspired and nature-inspired optimization methods are being proposed at any time in this fruitful area of research, it is expected that newer optimization techniques would also be tried in the near future in the automatic design of optimal type-2 fuzzy systems.

## References

[1] Z. Ahmad, H.M. Azamathulla, A.Ab. Ghani, ANFIS-based approach for estimation transverse mixing coefficient, IWA: Water Science & Technology 63 (2011) 1004–1009.
[2] L. Astudillo, P. Melin, O. Castillo, New optimization method base on a paradigm inspired by nature, Studies in Computational Intelligence 312 (2010) 277–283.
[3] H.M. Azamathulla, A.Ab. Ghani, N.A. Zakaria, ANFIS based approach for predicting maximum scour location of spillway, Water Management, ICE London 162 (2009) 399–407.
[4] H.M. Azamathulla, A.Ab. Ghani, An ANFIS-based approach for predicting the scour depth at culvert outlet, ASCE, Journal of Pipeline systems Engineering Practice 2 (2011) 35–40.
[5] H.M. Azamathulla, F.-C. Wu, Support vector machine approach to for longitudinal dispersion coefficients in streams, Applied Soft Computing 11 (2011) 2902–2905.
[6] M. Biglarbegian, W.W. Melek, J.M. Mendel, Design of novel interval type-2 fuzzy controllers for modular and reconfigurable robots: theory and experiments, IEEE Transactions on Industrial Electronics 58 (2011) 1371–1384.
[7] Z. Bingül, O. Karahan, A fuzzy logic controller tuned with PSO for 2 DOF robot trajectory control, Expert Systems with Applications 38 (2011) 1017–1031.
[8] J. Cao, P. Li, H. Liu, D. Brown, Adaptive fuzzy controller for vehicle active suspensions with particle swarm optimization, Proceedings of SPIE: The International Society of Optical Engineering (2008) 7129.
[9] O. Castillo, P. Melin, Soft Computing for Control of Non-Linear Dynamical Systems, Springer-Verlag, Heidelberg, Germany, 2001.
[10] O. Castillo, P. Melin, Soft Computing and Fractal Theory for Intelligent Manufacturing, Springer-Verlag, Heidelberg, Germany, 2003.
[11] O. Castillo, A.I. Martinez, A.C. Martinez, Evolutionary computing for topology optimization of type-2 fuzzy systems, Advances in Soft Computing 41 (2007) 63–75.
[12] O. Castillo, G. Huesca, F. Valdez, Evolutionary computing for topology optimization of type-2 fuzzy controllers, Studies in Fuzziness and Soft Computing 208 (2008) 163–178.
[13] O. Castillo, P. Melin, Type-2 Fuzzy Logic: Theory and Applications, Springer-Verlag, Heidelberg, Germany, 2008.
[14] O. Castillo, L.T. Aguilar, N.R. Cazarez-Castro, S. Cardenas, Systematic design of a stable type-2 fuzzy logic controller, Applied Soft Computing Journal 8 (2008) 1274–1279.
[15] O. Castillo, R. Martinez-Marroquin, P, Melin, F. Valdez, J. Soria, Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot, Information Sciences, in press.
[16] O. Castillo, P. Melin, A. Alanis, O. Montiel, R. Sepulveda, Optimization of interval type-2 fuzzy logic controllers using evolutionary algorithms, Journal of Soft Computing 15 (6) (2011) 1145–1160.
[17] J.R. Castro, O. Castillo, P. Melin, An interval type-2 fuzzy logic toolbox for control applications, in: Proceedings of FUZZ-IEEE 2007, London, pp. 1–6.
[18] J.R. Castro, O. Castillo, L.G. Martinez, Interval type-2 fuzzy logic toolbox, Engineering Letters 15 (1) (2007) 14.
[19] J.R. Castro, O. Castillo, P. Melin, L.G. Martinez, S. Escobar, I. Camacho, Building fuzzy inference systems with the interval type-2 fuzzy logic toolbox, Advances in Soft Computing 41 (2007) 53–62.
[20] J.R. Castro, O. Castillo, P. Melin, A. Rodriguez-Diaz, A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks, Information Sciences 179 (2009) 2175–2193.
[21] N.R. Cazarez-Castro, L.T. Aguilar, O. Castillo, Hybrid genetic-fuzzy optimization of a type-2 fuzzy logic controller, in: Proceedings of the 8th International Conference on Hybrid Intelligent Systems, HIS 2008, Barcelona, 2008, pp. 216–221.
[22] L. Cervantes, O. Castillo, Design of a fuzzy system for the longitudinal control of an F-14 airplane, Studies in Computational Intelligence 318 (2010) 213–224.
[23] F.C. Chang, H.C. Huang, A refactoring method for cache-efficient swarm intelligence algorithms, Information Sciences (doi:10.1016/j.ins.2010.02.025).

Please cite this article in press as: O. Castillo, P. Melin, A review on the design and optimization of interval type-2 fuzzy controllers, Appl. Soft Comput. J. (2012), doi:10.1016/j.asoc.2011.12.010

[24] H. Chaoui, W. Gueaieb, Type-2 fuzzy logic control of a flexible-joint manipulator, Journal of Intelligent and Robotic Systems: Theory and Applications 51 (2008) 159–186.
[25] O. Cordon, F. Gomide, F. Herrera, F. Hoffmann, L. Magdalena, Ten years of genetic fuzzy systems: current framework and new trends, Fuzzy Sets and Systems 141 (2004) 5–31.
[26] S. Coupland, R. John, New geometric inference techniques for type-2 fuzzy sets, International Journal of Approximate Reasoning 49 (2008) 198–211.
[27] T. Dereli, A. Baykasoglu, K. Altun, A. Durmusoglu, I.B. Turksen, Industrial applications of type-2 fuzzy sets and systems: a concise review, Computers in Industry 62 (2011) 125–137.
[28] A.P. Engelbrecht, Fundamentals of Computational Swarm Intelligence, John Wiley, England, 2005.
[29] M. Galluzzo, B. Cosenza, Adaptive type-2 fuzzy logic control of a bioreactor, Chemical Engineering Science 65 (2010) 4208–4221.
[30] H. Hagras, Hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots, IEEE Transactions on Fuzzy Systems 12 (2004) 524–539.
[31] M. Hsiao, T.H.S. Li, J.Z. Lee, C.H. Chao, S.H. Tsai, Design of interval type-2 fuzzy sliding-mode controller, Information Sciences 178 (2008) 1686–1716.
[32] C.H. Hsu, W.J. Shyr, K.H. Kuo, Optimizing multiple interference cancellations of linear phase array based on particle swarm optimization, Journal of Information Hiding and Multimedia Signal Processing 1 (2010) 292–300.
[33] H.C. Huang, C.M. Chu, J.S. Pan, The optimized copyright protection system with genetic watermarking, Soft Computing 13 (2009) 333–343.
[34] E.A. Jammeh, M. Fleury, C. Wagner, H. Hagras, M. Ghanbari, Interval type-2 fuzzy logic congestion control for video streaming across IP networks, IEEE Transactions on Fuzzy Systems 17 (2009) 1123–1142.
[35] C.-F. Juang, C.-H. Hsu, Reinforcement ant optimized fuzzy controller for mobile-robot wall-following control, IEEE Transactions on Industrial Electronics 56 (2009) 3931–3940.
[36] C.F. Juang, C.H. Hsu, Reinforcement interval type-2 fuzzy controller design by online rule generation and *Q*-value-aided ant colony optimization, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39 (2009) 1528–1542.
[37] C.F. Juang, C.H. Hsu, C.F. Chuang, Reinforcement self-organizing interval type-2 fuzzy system with ant colony optimization, in: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, San Antonio, 2009, pp. 771–776.
[38] H.G. Kaganami, S.K. Ali, B. Zou, Optimal approach for texture analysis and classification based on wavelet transform and neural network, Journal of Information Hiding and Multimedia Signal Processing 2 (2011) 33–40.
[39] W.-D. Kim, H.-J. Jang, S.-K. Oh, The design of optimized fuzzy cascade controller: focused on type-2 fuzzy controller and HFC-based genetic algorithms, Transactions of the Korean Institute of Electrical Engineers 59 (2010) 972–980.
[40] G.O. Koca, Z.H. Akpolat, M. Özdemir, Type-2 fuzzy sliding mode control of a four-bar mechanism, International Journal of Modelling and Simulation 31 (2011) 60–68.
[41] C.-H. Lee, Y.-C. Lin, An adaptive type-2 fuzzy neural controller for nonlinear uncertain systems, Control and Intelligent Systems 33 (2005) 13–25.
[42] C. Li, J. Yi, D. Zhao, Control of the TORA system using SIRMs based type-2 fuzzy logic, IEEE International Conference on Fuzzy Systems (2009) 694–699.
[43] O. Linda, M. Manic, Comparative analysis of type-1 and type-2 fuzzy control in context of learning behaviors for mobile robotics, in: IECON Proceedings (Industrial Electronics Conference), art. no. 5675521, 2010, pp. 1092–1098.
[44] R. Martinez, O. Castillo, L.T. Aguilar, Optimization with genetic algorithms of interval type-2 fuzzy logic controllers for an autonomous wheeled mobile robot: a comparison under different kinds of perturbations, in: Proceedings of the IEEE FUZZ Conference, 2008, paper #FS0225.
[45] R. Martinez, O. Castillo, L.T. Aguilar, Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms, Information Sciences 179 (2009) 2158–2174.
[46] R. Martinez, O. Castillo, L.T. Aguilar, A. Rodriguez, Optimization of type-2 fuzzy logic controllers using PSO applied to linear plants, Studies in Computational Intelligence 318 (2010) 181–193.
[47] R. Martinez, A. Rodriguez, O. Castillo, L.T. Aguilar, Type-2 fuzzy logic controllers optimization using genetic algorithms and particle swarm optimization, in: Proceedings of the IEEE International Conference on Granular Computing, GrC 2010, 2010, pp. 724–727.
[48] R. Martinez-Marroquin, O. Castillo, J. Soria, Parameter tuning of membership functions of a type-1 and type-2 fuzzy logic controller for an autonomous wheeled mobile robot using ant colony optimization, in: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, San Antonio, 2009, pp. 4770–4775.
[49] P. Melin, O. Castillo, A new method for adaptive model-based control of non-linear dynamic plants using a neuro-fuzzy-fractal approach, Journal of Soft Computing 5 (2001) 171–177.
[50] P. Melin, O. Castillo, A new method for adaptive model-based control of non-linear plants using type-2 fuzzy logic and neural networks, in: Proceedings of the IEEE FUZZ Conference, 2003, pp. 420–425.
[51] P. Melin, O. Castillo, A new method for adaptive control of non-linear plants using type-2 fuzzy logic and neural networks, International Journal of General Systems 33 (2004) 289–304.
[52] P. Melin, O. Castillo, An intelligent hybrid approach for industrial quality control combining neural networks, fuzzy logic and fractal theory, Information Sciences 177 (2007) 1543–1557.
[53] J.M. Mendel, Uncertainty, fuzzy logic, and signal processing, Signal Processing Journal 80 (2000) 913–933.
[54] O. Mendoza, P. Melin, O. Castillo, Interval type-2 fuzzy logic and modular neural networks for face recognition applications, Applied Soft Computing Journal 9 (2009) 1377–1387.
[55] S.M.A. Mohammadi, A.A. Gharaveisi, M. Mashinchi, An evolutionary tuning technique for type-2 fuzzy logic controller in a non-linear system under uncertainty, in: Proceedings of the 18th Iranian Conference on Electrical Engineering, ICEE 2010, pp. 610–616.
[56] S.-K. Oh, H.-J. Jang, W. Pedrycz, A comparative experimental study of type-1/type-2 fuzzy cascade controller based on genetic algorithms and particle swarm optimization, Expert Systems with Applications 38 (9) (2011) 11217–11229.
[57] K.J. Poornaselvan, T. Gireesh Kumar, V.P. Vijayan, Agent based ground flight control using type-2 fuzzy logic and hybrid ant colony optimization to a dynamic environment, in: Proceedings of the 1st International Conference on Emerging Trends in Engineering and Technology, ICETET 2008, 2008, pp. 343–348.
[58] K.M. Singh, Fuzzy rule based median filter for gray-scale images, Journal of Information Hiding and Multimedia Signal Processing 2 (2011) 108–122.
[59] J.T. Starczewski, Efficient triangular type-2 fuzzy logic systems, International Journal of Approximate Reasoning 50 (2009) 799–811.
[60] K.R. Sudha, R. Vijaya Santhi, Robust decentralized load frequency control of interconnected power system with generation rate constraint using type-2 fuzzy approach, International Journal of Electrical Power and Energy Systems 33 (2011) 699–707.
[61] M. Wagenknecht, K. Hartmann, Application of fuzzy sets of type 2 to the solution of fuzzy equations systems, Fuzzy Sets and Systems 25 (1988) 183–190.
[62] C. Wagner, H. Hagras, A genetic algorithm based architecture for evolving type-2 fuzzy logic controllers for real world autonomous mobile robots, in: Proceedings of the IEEE Conference on Fuzzy Systems, London, 2007.
[63] C. Wagner, H. Hagras, Evolving type-2 fuzzy logic controllers for autonomous mobile robots, Advances in Soft Computing 41 (2007) 16–25.
[64] C. Walker, E. Walker, Sets with type-2 operations, International Journal of Approximate Reasoning 50 (2009) 63–71.
[65] C.-H. Wang, C.-S. Cheng, T.-T. Lee, Dynamical optimal training for interval type-2 fuzzy neural network (T2FNN), IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 34 (3) (2004) 1462–1477.
[66] D. Wu, W.-W. Tan, A type-2 fuzzy logic controller for the liquid level process, in: Proceedings of the IEEE Conference on Fuzzy Systems, Budapest, 2004, pp. 953–958.
[67] D. Wu, W.-W. Tan, Genetic learning and performance evaluation of interval type-2 fuzzy logic controllers, Engineering Applications of Artificial Intelligence 19 (2006) 829–841.
[68] R.R. Yager, Fuzzy subsets of type II in decisions, Journal of Cybernetics 10 (1980) 137–159.
[69] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, Information Sciences 8 (1975) 43–80.
[70] M. Zaher, H. Hagras, Data generated type-2 fuzzy logic model for control of wind turbines, in: Proceedings of the 2010 10th International Conference on Intelligent Systems Design and Applications, ISDA'10, pp. 80–85.
[71] M.H.F. Zarandi, I.B. Turksen, O.T. Kasbi, Type-2 fuzzy modelling for desulphurization of steel process, Expert Systems with Applications 32 (2007) 157–171.

Please cite this article in press as: O. Castillo, P. Melin, A review on the design and optimization of interval type-2 fuzzy controllers, Appl. Soft Comput. J. (2012), doi:10.1016/j.asoc.2011.12.010