Performance Comparison of Efficient Type-Reduction Approaches for Interval Type-2 Fuzzy Logic Control

Zhikang Song, Jian Huang and Dongrui Wu
School of Automation, Huazhong University of Science and Technology, Wuhan, China
Email: zksong01@gmail.com, huang_jan@mail.hust.edu.cn, drwu@hust.edu.cn

***Abstract*—Although many articles have shown that the control performance of interval type-2 fuzzy logic controllers (IT2 FLCs) is better than type-1 FLCs, the computational cost of it is high, which makes it hard to develop in the real world. Our previous research has recommended six TR approaches for their efficiency, but which one has the best performance is still an unknown problem. In the paper, IT2 FLCs using these six TR approaches are optimized using genetic algorithms, and we compared their control performances based on a coupled tank apparatus for controlling the water level. Results show that there is no statistically significant difference among the control performances of these approaches.**

I. Introduction

Zadeh [16] proposed type-1 fuzzy set (T1 FS) theory in 1965 firstly, and it has been used in many fields successfully, such as data mining [8], time-series prediction [5], modeling and control [9], etc.

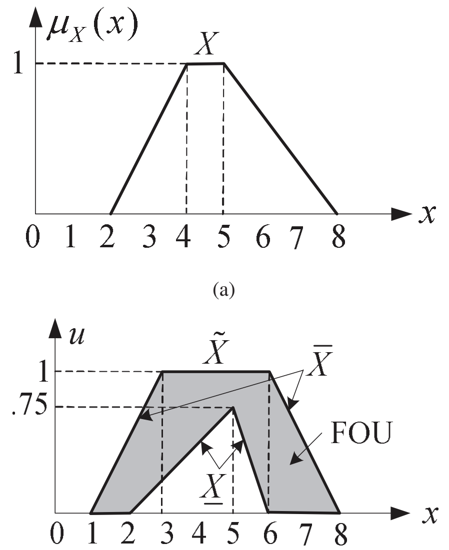

Fig. 1. Example of FS: (a) a T1 FS and (b) an IT2 FS.

Although T1 FSs has strong robustness than PID, its abilities in handling uncertainties' problems are limited [6]. The reason is simple, for T1 FS, its membership grade is certain for a crsip input. To solve this problem, Zadeh [17] proposed Type-2 FSs in 1975, whose membership grades themselves are T1 FS, but because of the high cost in computation, the Interval type-2 (IT2) FSs [6], whose secondary MFs are all equal to 1, are the most widely used from now.

An example of an FS, is shown in Fig. 1. It can be observed that for a T1 FS, its membership for each x is a crisp number [shown in Fig. 1(a)], for an IT2 FS, its membership is an interval for each x [shown in Fig. 1(b)]. For example, for the input 5, the membership of T1 FS is crisp number 1, that of IT2 is the interval [0, 1]. It can be also observed that the boundary of IT2 FS is determined by two T1 FSs, $\underline{X}$ and $\overline{X}$, which are called *lower membership function* (LMF) and *upper membership function* (UMF), respectively. The gray area called the *footprint of uncertainty* (FOU) is between $\underline{X}$ and $\overline{X}$.

IT2 FSs has one more degree of freedom (DOF) than T1 FSs, which indicated its potential in handling uncertainties' problems, so when the model is hard to built or exact MF is difficult to determine, IT2 FSs are good choice. To construct the MFs, we can depend on surveys or just use optimization algorithms, such as genetic algorithms.

The schematic diagram of an IT2 fuzzy logic controller(FLC) is shown in Fig. 2. Beside the FS are IT2 and the additional operation: type-reduction (TR), it is the same to T1 , the TR method is used to transform the IT2 FS into T1 FS, so that we can carry out the operation of defuzzification.

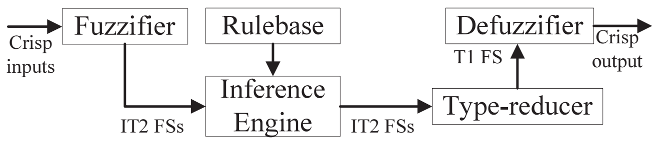

Fig. 2. An IT2 FLC.

Assuming that an IT2 FLC's rulebase has N rules with the following forms (if $\cdots$ then $\cdots$):

$$R^n: \quad \text{IF } x_1 \text{ is } \widetilde{X}_1^n \text{ and } \cdots \text{ and } x_I \text{ is } \widetilde{X}_I^n, \quad \text{THEN } y \text{ is } Y^n$$
$$n \subseteq [1, N]$$

where $\widetilde{X}_i^n$ $(i \subseteq [1, I])$ means the ith IT2 FSs, and the output of y is an interval, equals to $[\underline{y}^n, \overline{y}^n]$. For simplicity, we assume $\underline{y}^n = \overline{y}^n$ in many applications.

978-1-5386-3524-7/17/$31.00 ©2017 IEEE

Generally speaking, there are four steps in computing an IT2 FLS:

1) For each input x_i', compute its membership on each IT2 FS X_i^n, then get $[\mu_{\underline{X}_i^n}(x_i'), \mu_{\overline{X}_i^n}(x_i')]$, where $i \subseteq [1, I]$, $n \subseteq [1, N]$.
2) For the n^{th} rule, compute the firing interval $F^n(\mathbf{x}')$:

$$\begin{aligned} F^n(\mathbf{x}') = [&\mu_{\underline{X}_1^n}(x_1') \times \cdots \times \mu_{\underline{X}_I^n}(x_I'), \\ &\mu_{\overline{X}_1^n}(x_1') \times \cdots \times \mu_{\overline{X}_I^n}(x_I')] \\ \equiv [&\underline{f}^n, \overline{f}^n], \quad n \subseteq [1, N] \end{aligned} \tag{1}$$

3) According to $F^n(\mathbf{x}')$ we get in the second step and the rule consequents Y^n, and using TR to get T1 FS, then we get [6]:

$$Y_{cos}(\mathbf{x}') = \bigcup_{\substack{f^n \in F^n(\mathbf{x}') \\ y^n \in Y^n}} \frac{\sum_{n=1}^{N} f^n y^n}{\sum_{n=1}^{N} f^n} = [y_l, \ y_r] \tag{2}$$

It has also been shown that [6]:

$$\begin{aligned} y_l &= \min_{k \in [1, N-1]} \frac{\sum_{n=1}^{k} \overline{f}^n \underline{y}^n + \sum_{n=k+1}^{N} \underline{f}^n \underline{y}^n}{\sum_{n=1}^{k} \overline{f}^n + \sum_{n=k+1}^{N} \underline{f}^n} \\ &\equiv \frac{\sum_{n=1}^{L} \overline{f}^n \underline{y}^n + \sum_{n=L+1}^{N} \underline{f}^n \underline{y}^n}{\sum_{n=1}^{L} \overline{f}^n + \sum_{n=L+1}^{N} \underline{f}^n} \end{aligned} \tag{3}$$

$$\begin{aligned} y_r &= \max_{k \in [1, N-1]} \frac{\sum_{n=1}^{k} \underline{f}^n \overline{y}^n + \sum_{n=k+1}^{N} \overline{f}^n \overline{y}^n}{\sum_{n=1}^{k} \underline{f}^n + \sum_{n=k+1}^{N} \overline{f}^n} \\ &\equiv \frac{\sum_{n=1}^{R} \underline{f}^n \overline{y}^n + \sum_{n=R+1}^{N} \overline{f}^n \overline{y}^n}{\sum_{n=1}^{R} \underline{f}^n + \sum_{n=R+1}^{N} \overline{f}^n} \end{aligned} \tag{4}$$

where the L and R are called *switch points*, and determined by

$$\underline{y}^L \leq y_l \leq \underline{y}^{L+1} \tag{5}$$
$$\overline{y}^R \leq y_r \leq \overline{y}^{R+1} \tag{6}$$

and $\{\underline{y}^n\}$ and $\{\overline{y}^n\}$ have been sorted in ascending order, respectively. y_l and y_r can be computed using the Karnik-Mendel (KM) algorithms [6] shown in Table I.
4) Finally, we get the output of defuzification:

$$y = \frac{y_l + y_r}{2}. \tag{7}$$

As its name shows, the cost of iterative KM algorithms is high in computation, so many approaches have been proposed to alleviate that, which can be classified into three categories [10]:

1) *Enhancements to the KM algorithms*, these algorithms base on KM algorithm, making improvements on its initialization or search operations.
2) *Alternative TR methods*. Most of these TR methods have closed-form representations, this made them faster than the first category. But they are approximate algorithms to the KM algorithms.

TABLE I
THE KM ALGORITHMS.

Step	compute the y_l	compute the y_r
1	Initialize $f^n = \frac{\underline{f}^n + \overline{f}^n}{2}$ $y = \frac{\sum_{n=1}^N y f^n}{\sum_{n=1}^N f^n}$	Initialze $f^n = \frac{\underline{f}^n + \overline{f}^n}{2}$ $y = \frac{\sum_{n=1}^N \overline{y} f^n}{\sum_{n=1}^N f^n}$
2	Find $l \in [1, N-1]$ s.t $\underline{y}^L < y_l \leq \underline{y}^{L+1}$	Find $r \in [1, N-1]$ $\overline{y}^R < y_l \leq \overline{y}^{R+1}$
3	set $f^n = \begin{cases} \overline{f}^n, if & n \leq l \\ \underline{f}^n, if & n > l \end{cases}$ get $y' = \frac{\sum_{n=1}^N y^n f^n}{\sum_{n=1}^N f^n}$	set $f^n = \begin{cases} \underline{f}^n, if & n \leq r \\ \overline{f}^n, if & n > r \end{cases}$ get $y' = \frac{\sum_{n=1}^N \overline{y}^n f^n}{\sum_{n=1}^N f^n}$
4.	If $y' \neq y$, set $y = y'$ and go to Step 2. else set $y_l = y$ and $L = l$, stop.	If $y \neq y$,set $y = y'$ and go to Step 2. else set $y_r = y$ and $R = r$, stop.

3) *Simplified IT2 FLCs*, where a bit number of IT2 FSs are used in the important regions and in the most regions, we use T1 FSs.

We have given a comprehensive comparison on the computational cost of these different approaches [10], and recommend the enhanced opposite direction search (EODS) algorithm [4] and the enhanced iterative algorithm with stop condition (EIASC) algorithm [12] in the first category, and the Wu-Tan (WT) [14], Nie-Tan (NT) [7], Begian-Melek-Mendel (BMM) [1] and Greenfield-Chiclana-Coupland-John (GCCJ) [3] methods in the second category. However, despite their efficiency, which one has the best control performance is still an open question. This paper tries to answer that question by comparing their control performance using a coupled-tank water level control apparatus [13].

The remainder of this paper is organized as follows: Section II introduces the five recommended TR approaches in [10]. Section III presents the experimental setup and results. Section IV draws conclusions.

II. TR APPROACHES

This section introduces the five recommended TR approaches in [10]: EIASC, WT, NT, BMM, and GCCJ.

A. The EIASC Algorithm

The EODS algorithm and the EIASC algorithm are the two most efficient algorithms in the first category, and the EIASC algorithm is much easier to implement. So, in this paper we use EIASC [12]. However, because EODS and EIASC give identical outputs, our conclusions drawn on EIASC also apply to EODS.

The EIASC algorithm is shown in Table II. It makes two improvements to the original IASC algorithm [2]: 1) a new stopping criterion; and, 2) a better initialization for computing y_r. Note that in EIASC, $\{\underline{y}^n\}_{n \subseteq [1,N]}$ and $\{\overline{y}^n\}_{n \subseteq [1,N]}$ need to be arranged from small to large, respectively. All other alternative TR methods introduced next do not require this.

TABLE II
THE EIASC ALGORITHM [12].

Step	compute the y_l	compute the y_r
1.	Initialize $a=\sum_{n=1}^{N} \underline{y}^n \overline{f}^n$ $b=\sum_{n=1}^{N} \underline{f}^n$ $L=0$	Initialize $a=\sum_{n=1}^{N} \overline{y}^n \underline{f}^n$ $b=\sum_{n=1}^{N} \underline{f}^n$ $R=N$
2.	Compute $L=L+1$ $a=a+\underline{y}^L(\overline{f}^L-\underline{f}^L)$ $b=b+\overline{f}^L-\underline{f}^L$ $y_l=a/b$	Compute $a=a+\overline{y}^R(\overline{f}^R-\underline{f}^R)$ $b=b+\overline{f}^R-\underline{f}^R$ $y_r=a/b$ $R=R-1$
3.	If $y_l \leq y^{L+1}$, stop; else, go to Step 2.	If $y_r \geq y^R$, stop; else, go to Step 2.

B. The Wu-Tan (WT) Method

This method is proposed by Wu and Tan [14], it based on the idea: find the equivalent T1 membership grade $\mu_{X_i^n}(x_i)$ to replace each firing interval $[\mu_{\underline{X}_i^n}(x_i), \mu_{\overline{X}_i^n}(x_i)]$, i.e.,

$$\mu_{X_i^n}(x_i) = \mu_{\overline{X}_i^n}(x_i) - h_i^n(\mathbf{x})[\mu_{\overline{X}_i^n}(x_i) - \mu_{\underline{X}_i^n}(x_i)]$$

where $h_i^n(\mathbf{x})$ the function of X in the nth IT2 FS.

Since the $\mu_{X_i^n}(x_i)$ is a number not an interval, which leads to the (f^n) is also a number, the output of defuzzification can be computed as

$$y = \frac{\sum_{n=1}^{N} y^n f^n}{\sum_{n=1}^{N} f^n}.$$

C. The Nie-Tan (NT) Method

NT method [7] just uses $\underline{f}^n$ and $\overline{f}^n$ as the weights of y^n, as the following equation shows:

$$y = \frac{\sum_{n=1}^{N} y^n(\underline{f}^n + \overline{f}^n)}{\sum_{n=1}^{N}(\underline{f}^n + \overline{f}^n)}.$$

D. The Begian-Melek-Mendel (BMM) Method

The form of BMM method [1] is, i.e.,

$$y = \alpha\frac{\sum_{n=1}^{N} \underline{f}^n y^n}{\sum_{n=1}^{N} \underline{f}^n} + \beta\frac{\sum_{n=1}^{N} \overline{f}^n y^n}{\sum_{n=1}^{N} \overline{f}^n}. \quad (8)$$

where α and β are corresponding coefficients. From the equation, we can clearly know that the idea of it is to take charge of the message of two T1 FLCs (LMFs and UMFs), and give them different weights, which can be get by optimization algorithms.

E. The Greenfield-Chiclana-Coupland-John (GCCJ) Collapsing Method

GCCJ method [3] is similar to WT method, the idea are both to find the embedded T1 FS, but which may be not the same. In this paper, to avoid the complex computation, we use the fake representative embedded T1 FS, i.e.,

$$\mu_X(x) = \frac{\mu_{\underline{X}}(x) + \mu_{\overline{X}}(x)}{2}$$

Note that when all $h_i^n(\mathbf{x}) = 1$ in WT method, the GCCJ method is the same as WT method.

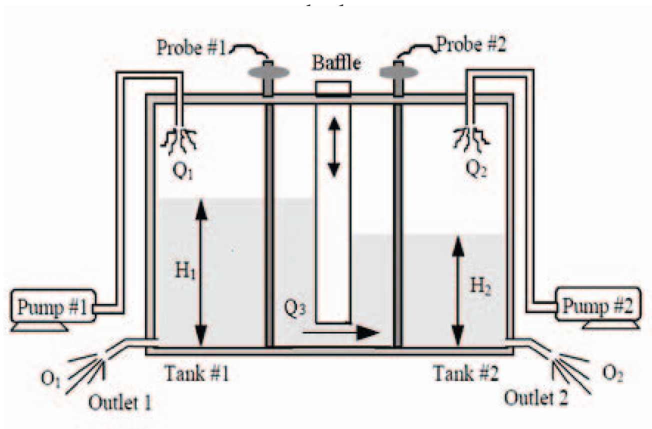

Fig. 3. The coupled-tank liquid level control apparatus.

III. EXPERIMENT AND RESULTS

This section introduces the experimental setup for comparing the control performances of the five TR approaches, and the results.

A. The Coupled-Tank Water Level Control Apparatus

A coupled-tank apparatus for water level controlling [13], shown in Fig. 3, was used in this paper to compare the control performances of the 5 TR approaches. The work process is that given a liquid level, then detect the real-time liquid level through the probe sensors, the controllers will feedback the signal to pumps for controlling the volumetric flow rate ($Q1$ and $Q2$).

The following equations describe the mathmatic dynamics of the coupled-tank apparatus:

$$A_1\frac{dH_1}{dt} = Q_1 - \alpha_1\sqrt{H_1} - \alpha_3\sqrt{H_1 - H_2} \quad (9)$$

$$A_2\frac{dH_2}{dt} = Q_2 - \alpha_2\sqrt{H_2} + \alpha_3\sqrt{H_1 - H_2} \quad (10)$$

where A_1, A_2, α_1, α_2, α_3 are the related constants, which can be computed by experiments; The water levels in Tanks 1 and 2 are H_1 and H_2; Q_1 and Q_2 are the flow rates (cm^3/s) of Pumps 1 and 2.

To make the experiment simple, we turn off the pump 2 to make the system become a one input one output system, then (10) becomes to:

$$A_2\frac{dH_2}{dt} = -\alpha_2\sqrt{H_2} + \alpha_3\sqrt{H_1 - H_2} \quad (11)$$

All IT2 FLCs with different TR approaches were tuned according to the simulated plant and 1s sampling period was used.

B. The Structure of IT2 FLCs

In the paper, all IT2 FLCs has two inputs: the feedback error e and its rate of change $\dot{e}$, and one output: the change of control signal $\dot{u}$, i.e., they implemented an incremental PI controller $\dot{u} = f(e, \dot{e})$. The rulebase is shown in Table III. Each input domain had three membership functions, and Gaussian

IT2 FSs were used for all inputs because our previous research has shown that they can avoid the control surface discontinuity problem [11]. Each Gaussian IT2 FS was defined by three parameters: a mean (m) and uncertain standard deviations $([\delta_1, \delta_2])$. Observe from Table III that there are 6 membership functions and 5 outputs, so each IT2 FLC had $3 \times 6 + 5 = 23$ parameters to tune.

TABLE III
RULEBASE OF THE IT2 FLCS.

$e \backslash \dot{e}$	$N_{\dot{e}}$	$Z_{\dot{e}}$	$P_{\dot{e}}$
N_e	$\dot{u}_1$	$\dot{u}_2$	$\dot{u}_3$
Z_e	$\dot{u}_2$	$\dot{u}_3$	$\dot{u}_4$
P_e	$\dot{u}_3$	$\dot{u}_4$	$\dot{u}_5$

C. The Optimization Procedure

A genetic algorithm (GA) is used to optimize the 23 parameters for each TR approach. Due to modelling uncertainties and the parameters we maybe only suitable for the model we designed, it is unavoidable that the performance will deteriorate in real model. To alleviate this problem, we exposed the IT2 FLCs to four kinds of uncertain model parameters (seen in Table IV) in GA tuning, as in [15]. The sum of the integral of the time-weighted absolute errors (ITAEs) is applied to evaluate the performance, defined in (12), and as a fitness function:

$$F = \sum_{i=1}^{4} \alpha_i \left[\sum_{j=1}^{N_i} j \cdot e_i(j) \right] \qquad (12)$$

where i is the ith plant, j is the jth sampling point, the difference between setpoint and real output is $e_i(j)$, α_i is the coefficient, which is related to the i^{th} plant, and the total sampling time we choose 200s, which means $N_i = 200$. Because the second plant's ITAE was usually larger than the rest three, we used $\alpha_2 = 1/3$ and $\alpha_1 = \alpha_3 = \alpha_4 = 1$.

Although theoretically GA is a global optimization algorithm, in practice due to randomness and limited running time, it may not always reach the global optimum. So, for each TR approach we repeated GA tuning 30 times and recorded the 30 best IT2 FLC configurations. Each GA had a population of 100 chromosomes, and the the maximum number of iterations is 500. We apply 0.1 as the mutation rate and 0.8 as the crossover.

D. Experiment Results

Fig. 4 shows the boxplot of the training performances of the 5 TR approaches. $h_i^n(\mathbf{x}) = 0.5$ was used in the WT method, so its results were identical to those of the GCCJ method. Observe that the training performances were very similar. Actually, an Analysis of Variance (ANOVA) showed that there was no statistically significant differences among the training performances of these TR approaches ($p = 0.5338$).

TABLE IV
THE FOUR PLANTS USED IN GA TUNING.

	I	II	III	IV
$A_1 = A_2$ (cm^2)	36.52	36.52	36.52	36.52
$\alpha_1 = \alpha_2$	5.6186	5.6186	5.6186	5.6186
α_3	10	10	10	8
Setpoint (cm)	$0 \to 15$	$0 \to 22.5 \to 7.5$	$0 \to 15$	$0 \to 15$
Transport delay (s)	0	0	2	0

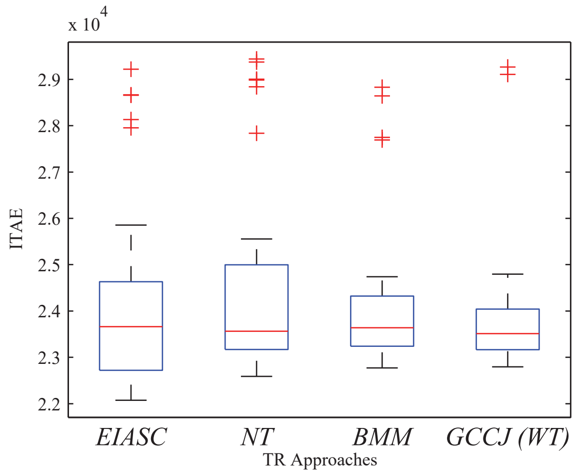

Fig. 4. Boxplot of the training performances of the 5 TR approaches.

Similar to [15], simulations were performed to test the abilities of the optimized IT2 FLCs in solving uncertainties. Fig. 5 shows the performances of the 5 TR approaches on the plant when $\alpha_3 = 8$ and a 1s delay was introduced. The performances were very similar. An ANOVA showed that there was no statistically significant differences among the testing performances of these TR approaches on this plant ($p = 0.3538$).

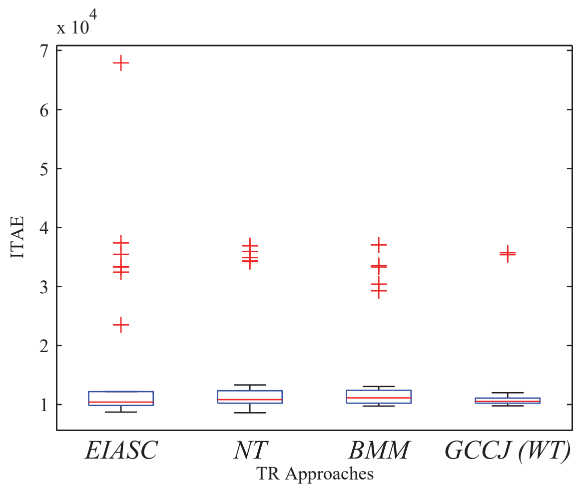

Fig. 5. Boxplot of the testing performances of the 5 TR approaches on the plant with $\alpha_3 = 8$ and a 1s delay.

Fig. 6 shows the performances of the 5 TR approaches on the plant when a 2s delay was introduced. Again, the performances were very similar. An ANOVA showed that there was no statistically significant differences among the testing performances of these TR approaches on this plant ($p = 0.3740$).

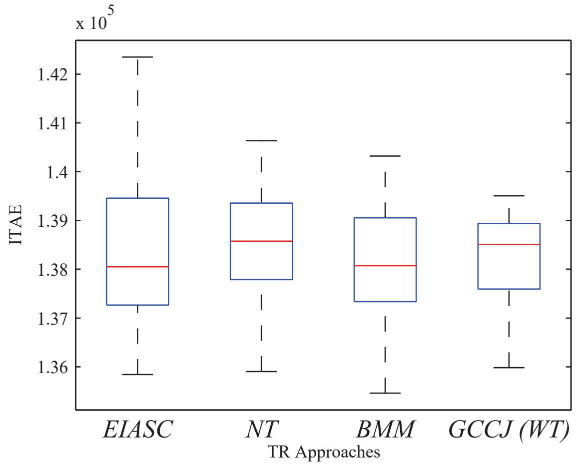

Fig. 6. Boxplot of the testing performances of the 5 TR approaches on the plant with a 2s delay.

In summary, both training and testing results showed that there were no statistically significant differences among the control performances of the 5 TR approaches (EIASC, WT, NT, BMM, and GCCJ). However, Table V, as well as Figs. 4-6, show that the ITAEs of GCCJ and WT may be more compact in both training and testing, i.e., their performance may be more consistent in different GA runs.

TABLE V
THE MEANS AND STANDARD DEVIATIONS OF THE ITAES OF THE 5 TR APPROACHES IN TRAINING AND TESTING.

	Training (Fig. 4)		Testing (Fig. 5)		Testing (Fig. 6)	
	mean (10^4)	std (10^3)	mean (10^4)	std (10^4)	mean (10^5)	std (10^3)
EIASC	2.43	2.10	1.67	1.34	1.38	1.65
NT	2.46	2.28	1.57	1.02	1.39	1.16
BMM	2.42	1.70	1.46	0.84	1.38	1.25
GCCJ (WT)	2.39	1.53	1.23	0.64	1.38	0.97

IV. Conclusions

Although IT2 FLCs have shown better performance than type-1 FLCs in many fields, the high cost of the KM TR algorithms in computation impedes its develop in real world. Our previous research has recommended six TR approaches for their efficiency, but which one has the best performance is still an unknown problem. In the paper, IT2 FLCs using these six TR approaches were optimized using GA, and their control performances were compared using a coupled-tank water-level control apparatus. Results showed that there is no statistically significant difference among the control performances of these approaches. So, any of them could be used in IT2 FLC design. However, we have yet to test their performances on a real plant to further verify this conclusion.

References

[1] M. Begian, W. Melek, and J. Mendel, "Stability analysis of type-2 fuzzy systems," in *Proc. IEEE Int'l Conf. on Fuzzy Systems*, Hong Kong, June 2008, pp. 947–953.

[2] K. Duran, H. Bernal, and M. Melgarejo, "Improved iterative algorithm for computing the generalized centroid of an interval type-2 fuzzy set," in *Proc. NAFIPS*, New York, May 2008, pp. 1–5.

[3] S. Greenfield, F. Chiclana, S. Coupland, and R. John, "The collapsing method of defuzzification for discretised interval type-2 fuzzy sets," *Information Sciences*, vol. 179, no. 13, pp. 2055–2069, 2008.

[4] H. Hu, Y. Wang, and Y. Cai, "Advantages of the enhanced opposite direction searching algorithm for computing the centroid of an interval type-2 fuzzy set," *Asian Journal of Control*, vol. 14, no. 6, pp. 1–9, 2012.

[5] S. S. Liao, T. H. Tang, and W.-Y. Liu, "Finding relevant sequences in time series containing crisp, interval, and fuzzy interval data," *IEEE Trans. on Systems, Man, and Cybernetics–B*, vol. 34, no. 5, pp. 2071–2079, 2004.

[6] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Upper Saddle River, NJ: Prentice-Hall, 2001.

[7] M. Nie and W. W. Tan, "Towards an efficient type-reduction method for interval type-2 fuzzy logic systems," in *Proc. IEEE Int'l Conf. on Fuzzy Systems*, Hong Kong, June 2008, pp. 1425–1432.

[8] W. Pedrycz, "Fuzzy set technology in knowledge discovery," *Fuzzy Sets and Systems*, vol. 98, pp. 279–290, 1998.

[9] L.-X. Wang, *A Course in Fuzzy Systems and Control*. Upper Saddle River, NJ: Prentice Hall, 1997.

[10] D. Wu, "Approaches for reducing the computational cost of interval type-2 fuzzy logic systems: Overview and comparisons," *IEEE Trans. on Fuzzy Systems*, vol. 21, no. 1, pp. 80–99, 2013.

[11] D. Wu and J. M. Mendel, "On the continuity of type-1 and interval type-2 fuzzy logic systems," *IEEE Trans. on Fuzzy Systems*, vol. 19, no. 1, pp. 179–192, 2011.

[12] D. Wu and M. Nie, "Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems," in *Proc. IEEE Int'l Conf. on Fuzzy Systems*, Taipei, Taiwan, June 2011, pp. 2131–2138.

[13] D. Wu and W. W. Tan, "A type-2 fuzzy logic controller for the liquid-level process," in *Proc. IEEE Int'l Conf. on Fuzzy Systems*, vol. 2, Budapest, Hungary, July 2004, pp. 953–958.

[14] ——, "Computationally efficient type-reduction strategies for a type-2 fuzzy logic controller," in *Proc. IEEE Int'l Conf. on Fuzzy Systems*, Reno, NV, May 2005, pp. 353–358.

[15] ——, "Genetic learning and performance evaluation of type-2 fuzzy logic controllers," *Engineering Applications of Artificial Intelligence*, vol. 19, no. 8, pp. 829–841, 2006.

[16] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

[17] ——, "The concept of a linguistic variable and its application to approximate reasoning-1," *Information Sciences*, vol. 8, pp. 199–249, 1975.