Design of Noniterative Algorithms for Takagi Sugeno Kang Type General Type-2 Fuzzy Logic Systems

Yang Chen (chenyanghanyun@163.com)
College of Science, Liaoning University of Technology

Chenxi Li
Liaoning University of Technology

Jiaxiu Yang
Liaoning University of Technology

Research Article

Keywords: general type-2 fuzzy logic systems, computational efficiency, center-of-sets type-reduction, alpha-planes, noniterative algorithms.

Posted Date: January 27th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1042114/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License
Design of noniterative algorithms for Takagi Sugeno Kang type general type-2 fuzzy logic systems

Yang Chen, Chenxi Li, Jiaxiu Yang

College of Science
Liaoning University of Technology
City of Jinzhou, 121001, China

Email: chenyanghanyun@163.com

Abstract: The paper performs the center-of-sets (COS) type-reduction (TR) and defuzzification for Takagi Sugeno Kang (TSK) type general type-2 fuzzy logic systems (GT2 FLSs) on the basis of the $\alpha$-planes expression of general type-2 fuzzy sets. Actually, comparing the popular Karnik-Mendel (KM) algorithms with other noniterative algorithms is an important question in T2 society. Here the modules of fuzzy inference, COS TR, and defuzzification for TSK type GT2 FLSs are discussed by means of noniterative Nagar-Bardini (NB) algorithms, Nie-Tan (NT) algorithms, and Begian-Melek-Mendel (BMM) algorithms. Simulation instances are constructed to illustrate the performances of three types of noniterative algorithms compared with the KM algorithms. It is proved that, the proposed noniterative algorithms can enhance the computational efficiencies significantly, which afford the potential application value for designers of GT2 FLSs.

Keywords: general type-2 fuzzy logic systems; computational efficiency; center-of-sets type-reduction; alpha-planes; noniterative algorithms.

1. Introduction

Interval type-2 fuzzy sets [1] can explain the uncertainties in membership functions (MFs). However, the secondary membership grades of interval type-2 fuzzy sets (IT2 FSs) are just equal
to 1, which must measure the uncertainty of MF uniformly. While the secondary membership grades of GT2 FSs lie between 0 and 1. Therefore, GT2 FSs can be regarded as higher-order uncertain fuzzy set models in contrast to IT2 FSs. Naturally, IT2 and GT2 FLSs use IT2 and GT2 FSs, respectively. As the design degrees of freedom increase, GT2 FLSs [2-5, 15] have advantages over IT2 FLSs [6-7] on many fields subject to uncertainty.

Generally, a GT2 FLS is constituted by five modules as: fuzzifier, fuzzy reasoning (inference), rules, TR and defuzzification (see the Fig. 1). Among which, the module of TR is especially important, which act as the role of varying the T2 fuzzy set to the T1 fuzzy set. Finally the defuzzification block maps the T1 fuzzy set to the output. In the past decades, the calculational costs of GT2 FLSs have been significantly reduced as the $\alpha$ -planes (or say z-Slices [8-10]) descriptions of GT2 FSs were put forward by some well-known researchers. Since then, GT2 FLSs are successfully applied to many fields as edge detection [11-12], intelligent fuzzy control [3, 5, 10], forecasting [4, 13-14], medical diagnosis [30] and so on.

![Fig. 1 Blocks of GT2 FLSs](image)

The centroid TR for IT2 FLSs is a very popular theoretical study approach. In the early days, the time-consuming Karnik and Mendel (KM) algorithms [16] were developed to complete the
centroid TR. Even so, the iterative property of KM algorithms made them difficult to apply in practical applications. Hence some noniterative algorithms were proposed gradually for perform the centroid TR, they are known as the Greenfield and Chiclana Collapsing Defuzzifier (GCCD [17]), Wu and Mendel uncertainty bound (UB [18]), Nagar and Bardini (NB) algorithms [19-21], Nie and Tan algorithms [22-24] and Begian and Melek and Mendel (BMM) algorithms [25-26]. In contrast to the centroid TR, studying the COS TR is more beneficial for designing IT2 and GT2 FLSs. Moreover, on the basis of alpha-planes representation of GT2 FSs, it is feasible to expand and improve the centroid TR of IT2 FLSs for performing the COS TR [27-29] of more complicated GT2 FLSs.

The paper expands the NB algorithms, NT algorithms and BMM algorithms to perform the COS TR for GT2 FLSs. Simulation experiments are constructed to illustrate the performances of three kinds of noniterative algorithms in contrast to the KM algorithms. The remainder of the paper is arranged as follows. Section two gives the TSK inference structure based GT2 FLS. Section three provides the proposed noniterative algorithms for performing the COS TR of GT2 FLSs. Six simulation experiments are provided in Section four to illustrate the performances of them. Finally Section five is the conclusions.

2. TSK GT2 FLSs

Similar to the inference structure of IT2 FLSs, GT2 FLSs are also divided into Mamdani type [13] and Takagi-Sugeno-Kang (TSK) type [2, 4, 14]. Here we only focus the TSK type. Consider a TSK type GT2 FLS with totally \( p \) inputs \( x_1 \in X_1, x_2 \in X_2, \ldots, x_p \in X_p \), and single output \( y \in Y \), which can be described by totally \( N \) fuzzy rules, in which the form \( l \)th rule is as:

\[
\text{If } x_1 \text{ is } F_1^l \text{ and } \cdots \text{ and } x_p \text{ is } F_p^l, \text{ then } y^l = c_0^l + c_1^l x_1 + \cdots + c_p^l x_p \quad (l = 1, 2, \cdots, N) \quad (1)
\]
in which \( \bar{F}_i^l (i = 1, \cdots, p) \) denotes the antecedent GT2 FS, and \( c_j^l (j = 0, 1, \cdots, p) \) represents the crisp consequent parameter. Here the input measurement is also chosen as the GT2 FS, furthermore, the GT2 FLSs are as the “A2–C0” type, i.e., the antecedent is the GT2 FS, and consequent is the crisp number. In addition, the structure of rules doesn’t not change when the systems vary from IT2 to GT2, and only the types of FSs are transformed.

Firstly, compute the consequents for each fuzzy rule, that is to say, \( \{ y'_a (x) \}_{i=1}^N \). Then renumber the \( \{ y' (x) \}_{i=1}^N \) as an ascending order \( \{ y'_a (x) \}_{i=1}^N \). Here the non-singleton fuzzifier is adopted to the firing interval \( F_a^l (x) \) for each rule under the corresponding \( \alpha \)-level as:

\[
F_a^l (x) = \left\{ \begin{array}{l}
\bar{f}_a^l (x) = T_{i=1}^p \bar{f}_a^l (x_i) = T_{i=1}^p \mu_{\overline{Q}_{a_i}} (x_{i\text{,max}}) = T_{i=1}^p [\mu_{\overline{Q}_{a_i}} (x_{i\text{,max}}) \wedge \overline{\mu}_{\overline{Q}_{a_i}} (x_{i\text{,max}})] \\
\bar{f}_a^l (x) = T_{i=1}^p \overline{f}_a^l (x_i) = T_{i=1}^p \mu_{\overline{Q}_{a_i}} (x_{i\text{,max}}) = T_{i=1}^p [\mu_{\overline{Q}_{a_i}} (x_{i\text{,max}}) \wedge \overline{\mu}_{\overline{Q}_{a_i}} (x_{i\text{,max}})]
\end{array} \right.
\]

(2)

here we reorder the \( F_a^l (x) \) as the order of \( \{ y'_a (x) \}_{i=1}^N \) and \( x_{i\text{,max}} \) are the \( x_i \) values which are in relation with \( \sup_{x_i} \mu_{\overline{Q}_{a_i}} (x_i) \) and \( \sup_{x_i} \overline{\mu}_{\overline{Q}_{a_i}} (x_i) \), respectively.

The output of TSK type GT2 FLSs under the \( \alpha \)-level can be calculated as:

\[
Y_{\text{FSK}, \alpha} = \alpha \left[ \frac{\sum_{i=1}^N F_a^l (x) y'_a^l (x)}{\sum_{i=1}^M F_a^l (x)} \right] = \alpha \left[ y_{l,a,x} (x), y_{r,a,x} (x) \right]
\]

(3)

where \( y_{l,a,x} (x) \) and \( y_{r,a,x} (x) \) can be computed by the KM types of TR algorithms as:

\[
y_{l,a,x} (x) = \min_{f_a^l (x) \neq T_{i=1}^p \overline{f}_a^l (x_i) \neq 0} \frac{\sum_{i=1}^N f_a^l (x) y'_a^l (x)}{\sum_{i=1}^N f_a^l (x)}
\]

(4)

and

\[
y_{r,a,x} (x) = \max_{f_a^l (x) \neq T_{i=1}^p \overline{f}_a^l (x_i) \neq 0} \frac{\sum_{i=1}^N f_a^l (x) y'_a^l (x)}{\sum_{i=1}^N f_a^l (x)}
\]

(5)
Let the whole number of effective $\alpha$-planes be $k$, that is to say, the $\alpha$ is equally divided into: $\alpha = \alpha_1, \alpha_2, \cdots, \alpha_k$. The final output of TSK type GT2 FLSs can be got as:

$$y_{TSK}(x) = \frac{\sum_{i=1}^{k} \alpha_i \{[y_{l,a_i}(x) + y_{r,a_i}(x)]/2\}}{\sum_{i=1}^{k} \alpha_i}$$

(6)

here the equation (6) was first put forward by Wagner [10], which can be called as the endpoints average defuzzification method. Although this approach can get the defuzzified value by a relatively simple way, which also need to compute the $k$ $Y_{TSK,a}$ values according to the corresponding $\alpha$.

3. Three kinds of noniterative algorithms

In this section, we obtain the output for TSK type GT2 FLSs according to the NB algorithms, NT algorithms and BMM algorithms.

3.1. NB algorithms

Recent studies show that T2 FLSs on the basis of NB algorithms [19] own superior capability in face of environment uncertainties and external disturbances. By means of the fuzzy reasoning [31], let the output COS type-reduced set of TSK type GT2 FLSs at the $\alpha$ level be an interval, that is to say, $y_{TSK,a} = [y_{l,a}, y_{r,a}]$. Then the two end points of interval can be calculated as:

$$y_{l,a} = \frac{\sum_{l=1}^{N} f_{a,l} y_{a,l}^l}{\sum_{l=1}^{N} f_{a,l}}$$  

(7)

and

$$y_{r,a} = \frac{\sum_{l=1}^{N} -f_{a,l} y_{a,l}^l}{\sum_{l=1}^{N} f_{a,l}}$$  

(8)
At the related $\alpha$-level, the COS defuzzified value can be obtained as:

$$y_{NB,\alpha} = \frac{1}{2} \cdot (y_{L,\alpha} + y_{r,\alpha}).$$

(9)

Aggregating all the $y_{NB,\alpha}$ to get the COS type-reduced set, i.e.,

$$Y_{NB} = \sup_{\alpha \in [0,1]} \alpha / Y_{NB,\alpha}.$$  

(10)

The final output of TSK type GT2 FLSs based on NB algorithms can be obtained as the form in equation (6), i.e.,

$$y_{NB} = \frac{\sum_{i=1}^{k} \alpha_i y_{NB,\alpha}}{\sum_{i=1}^{k} \alpha_i}.$$  

(11)

As a matter of fact, the simple closed form NB algorithms get the output from the linear combination of two different type-1 FLSs: one depends on the upper membership functions of type-2 fuzzy sets, and the other relies on the lower membership functions of T2 FSs.

### 3.2. NT algorithms

For the centroid type-reduction, the latest researches prove that the continuous Nie and Tan (CNT) algorithms [22] are actually an exact approach. Moreover, the sampling-based NT algorithms [32-33] can precisely approach the CNT algorithm. Here the COS type-reduction for GT2 FLSs is studied based on the NT algorithms. At the corresponding $\alpha$-level, the COS defuzzified value can be computed by the NT algorithms as:

$$y_{NT,\alpha} = \frac{\sum_{l=1}^{N} y_{l}^f \left( f_{l}^0 + \overline{f}_{l}^0 \right)}{\sum_{l=1}^{N} \left( f_{l}^0 + \overline{f}_{l}^0 \right)}.$$  

(12)

Aggregating all the $y_{NT,\alpha}$ to get the COS type-reduced set, i.e.,

$$Y_{NT} = \sup_{\alpha \in [0,1]} \alpha / Y_{NT,\alpha}.$$  

(13)
Finally the output of TSK type GT2 FLSs can be calculated as:

\[ y_{NT} = \frac{\left( \sum_{i=1}^{k} \alpha_i y_{NT,a} \right)}{\left( \sum_{i=1}^{k} \alpha_i \right)} \]. \hspace{1cm} (14)

Actually, just choose the average of upper and lower firing intervals, the closed form of NT algorithms can be got. In addition, this type of closed form of algorithm can perform the type-reduction and defuzzification simultaneously.

### 3.3. BMM algorithms

IT2 FLSs based on BMM algorithms [25-26] have preferable robustness and stability in contrast to T1 FLSs. Now we study the COS type-reduction of TSK type GT2 FLSs based on BMM algorithms. At the corresponding \(\alpha\)-level, the COS defuzzified value can be calculated by the BMM algorithms as:

\[ y_{BMM,\alpha} = a_{BMM} \frac{\sum_{l=1}^{N} y_{\alpha}^l f_{\alpha}^l}{\sum_{l=1}^{N} f_{\alpha}^l} + b_{BMM} \frac{\sum_{l=1}^{N} y_{\alpha}^l \tilde{f}_{\alpha}^l}{\sum_{l=1}^{N} \tilde{f}_{\alpha}^l} \]. \hspace{1cm} (15)

where \(a_{BMM}\) and \(b_{BMM}\) are two adjustable coefficients. Based on two T1 FLSs, among which, one depends on the upper membership functions, and the other depends on the lower membership functions.

In addition, the BMM algorithms are actually more general form of above two kinds of algorithms. Here we provide the explanations as follows.

See the equations (7)-(9), and (15), it can be easily found that the BMM algorithms and NB algorithms become the same as the coefficients are chosen as \(a_{BMM} = a_{NB} = \frac{1}{2}\), and \(b_{BMM} = b_{NB} = \frac{1}{2}\). In like manners, the equation (12) can be transformed to:
Then it can be reexpressed as:

\[ Y_{NT,a} = a_{NT} \frac{\sum_{l=1}^{N} \gamma_{a}^l f_{-a}^l}{\sum_{l=1}^{N} f_{-a}^l} + b_{NT} \frac{\sum_{l=1}^{N} \gamma_{a}^l \bar{f}_{a}^l}{\sum_{l=1}^{N} \bar{f}_{a}^l} \]  

(17)

in which

\[ a_{NT} = \frac{\left( \sum_{l=1}^{N} f_{-a}^l \right)^{1/2}}{\left( \sum_{l=1}^{N} (f_{-a}^l + \bar{f}_{a}^l) \right)^{1/2}} \]  

(18)

and

\[ b_{NT} = \frac{\left( \sum_{l=1}^{N} \bar{f}_{a}^l \right)^{1/2}}{\left( \sum_{l=1}^{N} (f_{-a}^l + \bar{f}_{a}^l) \right)^{1/2}}. \]  

(19)

Observing the equations (15) and (17), we can easily get that the BMM algorithms and NT algorithms prove to be the same as the coefficients are chosen as \( a_{BMM} = a_{NT} \), and \( b_{BMM} = b_{NT} \).

Based on the above mentioned analysis, the conclusion is that the BMM algorithms are more general form of above two kinds of algorithms, i.e., the latter are two special forms of the former.

4. Simulation experiments

Here we provide six simulation experiments to show the performances of three types of noniterative algorithms to complete the COS TR for TSK type GT2 FLSs. In these experiments, we divide the \( \alpha \) equally into \( \Delta \) effective values as: \( 0,1/\Delta,2/\Delta,\cdots,(\Delta-1)/\Delta,1 \).

Furthermore, suppose that \( \Delta \) be varied from one to a hundred with the step size of one. As \( \Delta = 100 \) (the maximum number), the COS type-reduced sets calculated by noniterative algorithms and KM algorithms are studied. In addition, when the number of valid alpha-planes \( \Delta \) varies...
from one to a hundred with the step size of one, the COS defuzzified values are also investigated.

For the first two examples, let each fuzzy rule be characterized by four antecedents GT2 FSs and one consequent GT2 FSs. Moreover, every antecedent can be characterized by two GT2 FSs. Therefore, there exists totally 16 fuzzy rules for TSK type GT2 FLSs. Let the \( sth \) fuzzy rule \( \tilde{R}^s \) be as:

If \( x_i \) is \( \tilde{F}_i^s \) and \( \cdots \) and \( x_4 \) is \( \tilde{F}_4^s \), then \( Y^s = C_0 + \sum_{i=1}^{4} C_i^s x_i \) \hspace{1cm} (20)

where \( \tilde{F}_i^s \) (\( i = 1, 2, \cdots, 4; s = 1, 2, \cdots, 16 \)) is the antecedent GT2 FS, \( C_i^s \) (\( i = 0, 1, \cdots, 4; s = 1, 2, \cdots, 16 \)) is the parameter for consequent, in which \( C_i^s = [c_i^s - s_i^s, c_i^s + s_i^s] \).

Fig. 2. Shape of footprint of uncertainty (FOU) for Gaussian primary membership function with uncertain standard deviation

**Simulation example one:** For each fuzzy rule, the primary MF of GT2 FS is selected as the
Gaussian primary MF type, observe the form in Fig. 2, i.e., the MF expression is as:

\[
\mu_i^s(x_i) = \exp\left[-\frac{1}{2} \left(\frac{x_i - m_i^s}{\sigma_i^s}\right)^2\right] \quad (i = 1, \ldots, 4; s = 1, \ldots, 16) \tag{21}
\]

where \(\sigma_i^s \in [\sigma_{i1}^s, \sigma_{i2}^s]\). While the secondary MF of antecedent is selected as the triangular type MF, i.e.,

\[
Apex = u_i(x) + 0.2[u_2(x) - u_i(x)] \tag{22}
\]

where \(u_i(x)\) and \(u_2(x)\) represent the lower and upper bounds of FOU, respectively.

Here we choose the lower and upper uncertain standard deviation, and mean for Gaussian antecedents primary MF as:

\[
\sigma_{i1}^s = 2.2 + \text{rand}(4,16) \tag{23}
\]
\[
\sigma_{i2}^s = \sigma_{i1}^s + \text{rand}(4,16) \tag{24}
\]
\[
m_i^s = 1.8 + 2*\text{rand}(4,16). \tag{25}
\]

Furthermore, the parameters for consequents \(c_i^s\) are selected as follows:

\[
c_i^s = \text{rand}(1,4), \quad s_i^s = \text{rand}(1,4) \quad (i = 0,1,\ldots,4) \tag{26}
\]

For the proposed TSK type GT2 FLSs, the input measurement set is as:

\[
x = 7*\text{rand}(16,4). \tag{27}
\]

**Simulation example two:** For each fuzzy rule, the primary MF of GT2 FS is chosen as the Gaussian primary MF with uncertain mean, observe the form in Figure 3, that is to say, the MF expression is as:

\[
\mu_i^s(x_i) = \exp\left[-\frac{1}{2} \left(\frac{x_i - m_i^s}{\sigma_i^s}\right)^2\right] (i = 1, \ldots, 4; s = 1, \ldots, 16) \tag{28}
\]

where \(m_i^s \in [m_{i1}^s, m_{i2}^s]\). The secondary MF of antecedent is selected as the trapezoidal type MF.
Fig. 3. Shape of FOU for Gaussian primary membership function with uncertain mean

i.e.,

Left top end point: \( L(x) = u_1(x) \) \hspace{1cm} (29)

Right top end point: \( R(x) = u_2(x) - \frac{3}{5}[u_2(x) - u_1(x)] \). \hspace{1cm} (30)

Then the parameters for antecedent are as:

\[ m_{i1}^s = 1.7 + \text{rand}(4,16) \] \hspace{1cm} (31)

\[ m_{i2}^s = m_{i1}^s + 2 \times \text{rand}(4,16) \] \hspace{1cm} (32)

\[ \sigma_i^s = 2.3 + \text{rand}(4,16) \] \hspace{1cm} (33)

Moreover, the parameters for consequents and input measurement are chosen as the same as in equations (26), and (27), respectively.

In the next simulation examples three and four, the forms of all parameters of TSK type GT2 FLSs are chosen the same as in simulation examples one and two, respectively. Even so, the total number of antecedents of each rule in these two examples is chosen as 5, therefore, these exists \( 2^5 \), i.e., 32 fuzzy rules fuzzy rules for TSK type GT2 FLSs. In the last two examples, we choose the total number of antecedents for each rule as 6, and the whole number of rules in TSK type
GT2 FLSs turn out to be $2^6$, i.e., 64.

Here both the quantitative and qualitative studies are done for three kinds of noniterative algorithms. For the more generalized BMM algorithms, one adjustable coefficient $a_{BMM}$ is selected as the mean of a thousand random numbers distributed on [0, 1], and the other one $b_{BMM} = 1 - a_{BMM}$. As for these above six examples, the COS type-reduced sets obtained by the proposed noniterative algorithms and the popular KM algorithms can be seen in Figure 4.
Moreover, as the number effective alpha-planes $\Delta = 1:1:100$, the defuzzified values of TSK type GT2 FLSs calculated by the proposed noniterative algorithms and the popular KM algorithms are shown in Fig. 5.
Fig. 5. COS defuzzified values, (a) example one; (b) example two; (c) example three; (d) example four; (e) example five; and (f) example six

Here we investigate the specific computational times for four types of algorithms. The hardware platform for programming is a dual core CPU dell desktop, and running with E5300@2.60 GHz, 2.00 GB memory and Windows XP. The software platform is the Matlab 2013a. For the sake of measuring the efficiencies of these types of algorithms, here we show the computational times of COS type-reduced sets and defuzzified values, and they are provide in Table1 and Table 2, respectively. Here the time unit is chosen as the second (s), while the last column denotes the average for these six examples, and the last line represents the defined time.
Table 1 Comparisons of calculating the COS type-reduced sets

| Num      | KM   | NB   | NT   | BMM  | TRR<sub>NB</sub> (%) | TRR<sub>NT</sub> (%) | TRR<sub>BMM</sub> (%) |
|----------|------|------|------|------|----------------------|----------------------|------------------------|
| Example one | 0.029870 | 0.019284 | 0.018623 | 0.022588 | 35.44                  | 37.65                  | 24.38                  |
| Example two | 0.047942 | 0.020114 | 0.016418 | 0.021194 | 58.05                  | 65.75                  | 55.79                  |
| Example three | 0.103220 | 0.048279 | 0.045325 | 0.053030 | 53.23                  | 56.09                  | 48.62                  |
| Example four | 0.076932 | 0.037100 | 0.040606 | 0.052199 | 51.78                  | 47.22                  | 32.15                  |
| Example five | 0.255190 | 0.114000 | 0.108771 | 0.159395 | 55.33                  | 57.38                  | 37.54                  |
| Example six  | 0.230725 | 0.097911 | 0.099250 | 0.150540 | 57.56                  | 56.98                  | 34.74                  |
| Average     | 0.123980 | 0.056115 | 0.054832 | 0.076491 | 54.74                  | 55.77                  | 38.30                  |

Table 2 Comparisons for computing the COS defuzzified values

| Num      | KM   | NB   | NT   | BMM  | TRR<sub>NB</sub> (%) | TRR<sub>NT</sub> (%) | TRR<sub>BMM</sub> (%) |
|----------|------|------|------|------|----------------------|----------------------|------------------------|
| Example one | 1.553890 | 0.894616 | 1.023340 | 1.038265 | 42.43                  | 34.14                  | 33.18                  |
| Example two | 1.856522 | 1.036907 | 1.023957 | 1.098692 | 44.15                  | 44.85                  | 40.82                  |
| Example three | 3.334768 | 2.094313 | 2.029514 | 2.109151 | 37.20                  | 39.14                  | 36.75                  |
| Example four | 3.776032 | 2.119719 | 2.099024 | 2.126635 | 43.86                  | 44.41                  | 43.68                  |
| Example five | 12.04636 | 5.320790 | 5.576742 | 7.675886 | 55.83                  | 53.71                  | 35.28                  |
| Example six  | 8.842907 | 4.905219 | 5.841277 | 6.123291 | 44.53                  | 33.94                  | 30.75                  |
| Average     | 5.235126 | 2.728594 | 2.932309 | 3.361987 | 47.88                  | 43.99                  | 35.78                  |

reducing rate (TRR) of NB, NT, and BMM noniterative algorithms compared with the KM algorithms, respectively. Here the TRR can be defined as:

\[
\text{TRR}_{NB,NT,BMM} = (t_{KM} - t_{NB,NT,BMM}) / t_{KM} \times 100\%
\]  \hspace{1cm} (34)

in which \( t_{KM} \) is the calculation time of KM algorithms, and \( t_{NB,NT,BMM} \) is the calculation time for NB, NT, and BMM algorithms.

Seeing the Figures 4-5 and the Tables 1-2, for these six examples, we can easily obtain the
following quantitative and qualitative analysis conclusions:

1) As for the COS type-reduced sets of TSK type GT2 FLSs, the calculational results of NB, NT and BMM noniterative algorithms always lie between the left and right COS type-reduced membership functions computed by the KM algorithms (observe the Figure 4).

2) As for the COS defuzzified values of TSK type GT2 FLSs, the computational results of four kinds of algorithms are not too much different. Moreover, the results of four types of algorithms are almost completely the same in example three, and the results of noniterative algorithms are almost the same in every example.

3) Compared with the KM algorithms, in these six examples, the proposed noniterative algorithms get the maximum time reducing rate (TRR) as 58.05%, 65.75% and 55.79% for obtaining the COS type-reduced sets. In addition, the proposed noniterative algorithms obtain the maximum TRR as 55.83%, 53.71% and 43.68% for calculating the COS defuzzified values.

4) As for the calculation times of COS type-reduced sets, the proposed noniterative algorithms obtain the average TRR as 54.74%, 55.77% and 38.30%; while for the calculation times of COS defuzzified values, the proposed noniterative algorithms obtain the average TRR as 47.88%, 43.99% and 35.78%.

Six computer simulation examples about calculating the COS type-reduced sets and defuzzified values of TSK type GT2 FLSs all show the effectiveness of proposed three kinds of discrete noniterative algorithms. Compared with the popular KM algorithms, the calculational times of noniterative algorithms are greatly reduced, that is to say, the latter can improve the calculational efficiency a lot. Therefore, it is reasonable to believe that we should adopt noniterative algorithms to investigate the COS type-reduction of GT2 FLSs.
5. Conclusion

TSK type GT2 FLSs on the basis of three kinds of noniterative are proposed in this paper. We also discuss the modules of inference, COS TR and defuzzification for the GT2 FLSs. As for calculating the COS type-reduced sets and defuzzified values, six experiments are given to show the effective of proposed three types of noniterative algorithms in contrast to the KM algorithms. As the computational times of former are significantly less than the latter, which can afford the potential values for adopting them in T2 FLSs to deal with uncertain environments.

In the next work, adopting both the iterative and noniterative algorithms for studying the centroid and COS type-reduction [16-26, 28-29, 32-35] of T2 FLSs will be further studied. Furthermore, seeking for the optimal antecedents and fuzzy rules for T2 FLSs on the basis of intelligent algorithms [3-5, 35-39] will also be studied. In conclusion, our studies will be concentrated on designing and applying T2 FLSs [40-45] in fields like forecasting, control, identification and so forth.

Acknowledgment. The paper is partially supported by the National Natural Science Foundation of China (NSFC) (61973146), the Doctoral Start-up Foundation of Liaoning Province (2021-BS-258), and the Talent Foundation of Liaoning University of Technology (xr2020002). The author is thankful to the famous scholar J. M. Mendel, who has provided many invaluable suggestions.

Funding. The paper is partially supported by the National Natural Science Foundation of China (NSFC) (61973146), the Doctoral Start-up Foundation of Liaoning Province (2021-BS-258), and the Talent Foundation of Liaoning University of Technology (xr2020002).
Compliance with ethical standards

Conflict of interest No conflict exists. The author declares that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] D. R. Wu, J. M. Mendel. Uncertainty measures for interval type-2 fuzzy sets. Information Sciences, 2007, 177(23): 5378-5393.

[2] J. M. Mendel. General type-2 fuzzy logic systems made simple: a tutorial. IEEE Transactions on Fuzzy Systems, 2014, 22(5): 1162-1182.

[3] O. Castillo, L. Amador-Angulo, J. R. Castro, M. Garcia-Valde. A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems. Information Sciences, 2016, 354(c): 257-274.

[4] Y. Chen, D. Z. Wang, W. Ning. Forecasting by TSK general type-2 fuzzy logic systems optimized with genetic algorithms. Optimal Control Applications & Methods, 2018, 39(1): 393-409.

[5] M. A. Sanchez, O. Castillo, J. R. Castro. Generalized type-2 fuzzy systems for controlling a mobile robot and a performance comparison with interval type-2 and type-1 fuzzy systems. Expert Systems with Applications, 2015, 42(14): 5904-5914.

[6] J. M. Mendel. Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Englewood Cliffs, NJ, USA: Prentice-Hall, 2001.

[7] D. R. Wu, J. M. Mendel. Recommendations on designing practical interval type-2 fuzzy logic
systems. Engineering Applications of Artificial Intelligence, 2019, 85: 182-193.

[8] F. Liu. An efficient centroid type reduction strategy for general type-2 fuzzy logic system. Information Sciences, 2008, 178(9): 2224-2236.

[9] J. M. Mendel, F. Liu, D. Zhai. Alpha-plane representation for type-2 fuzzy sets: theory and applications. IEEE Transactions on Fuzzy Systems, 2009, 17(5): 1189-1207.

[10] C. Wagner, H. Hagras. Towards general type-2 fuzzy logic systems based on zSlices. IEEE Transactions on Fuzzy Systems, 2010, 18(4): 637-660.

[11] C. I. Gonzalez, P. Melin, J. R. Castro, et al. An improved sobel edge detection method based on generalized type-2 fuzzy logic. Soft Computing, 2016, 20(2): 773-784.

[12] P. Melin, C. I. Gonzalez, J. R. Castro, et al. Edge-detection method for image processing based on generalized type-2 fuzzy logic. IEEE Transactions on Fuzzy Systems, 2014, 22(6): 1515-1525.

[13] Y. Chen, D. Z. Wang. Forecasting by designing Mamdani general type-2 fuzzy logic systems optimized with quantum particle swarm optimization algorithms. Transactions of the Institute of Measurement and Control, 2019, 41(10): 2886-2896.

[14] Y. Chen, D. Z. Wang. Forecasting by general type-2 fuzzy logic systems optimized with QPSO algorithms. International Journal of Control, Automation and Systems, 2017, 15(6): 2950-2958.

[15] D. Y. Zhai, J. M. Mendel. Uncertainty measures for general type-2 fuzzy sets. Information Sciences, 2011, 181(3): 503-518.

[16] J. M. Mendel. On KM algorithms for solving type-2 fuzzy set problems. IEEE Transactions on Fuzzy Systems, 2013, 21(3): 426-446.
[17] S. Greenfield, F. Chiclana, S. Coupland, R. John. The collapsing method of defuzzification for discretised interval type-2 fuzzy sets. Information Sciences, 2009, 179(13): 2055-2069.

[18] H. Wu, J. M. Mendel. Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems. IEEE Transactions on Fuzzy Systems, 2002, 10(5): 622-639.

[19] A. M. El-Nagar, M. El-Bardini. Simplified interval type-2 fuzzy logic system based on new type-reduction. Journal of Intelligent & Fuzzy Systems, 2014, 27(4): 1999-2010.

[20] Y. Chen. Study on weighted Nagar-Bardini algorithms for centroid type-reduction of general type-2 fuzzy logic systems. Journal of Intelligent & Fuzzy Systems, 2019, 37(5): 6527-6544.

[21] Y. Chen. Study on weighted Nagar-Bardini algorithms for centroid type-reduction of interval type-2 fuzzy logic systems. Journal of Intelligent & Fuzzy Systems, 2018, 34(4): 2417-2428.

[22] J. W. Li, R. John, S. Coupland, G. Kendall. On Nie-Tan operator and type-reduction of interval type-2 fuzzy sets. IEEE Transactions on Fuzzy Systems, 2018, 26(2): 1036-1039.

[23] J. M. Mendel, X. W. Liu. Simplified interval type-2 fuzzy logic systems. IEEE Transactions on Fuzzy Systems, 2013, 21(6): 1056-1069.

[24] Y. Chen, D. Z. Wang. Study on centroid type-reduction of general type-2 fuzzy logic systems with weighted Nie-Tan algorithms. Soft Computing, 2018, 22(22): 7659-7678.

[25] M. Biglarbegian, W. W. Melek, J. M. Mendel. On the robustness of type-1 and interval type-2 fuzzy logic systems in modeling. Information Sciences, 2011, 181(7): 1325-1347.

[26] M. Biglarbegian, W. W. Melek, J. M. Mendel. On the stability of interval type-2 TSK fuzzy logic systems. IEEE Transactions on Systems, Man and Cybernetics. Part B, Cybernetics, 2010, 40(3): 798-818.

[27] E. Ontiveros, P. Melin, O. Castillo. Higher order α-planes integration: a new approach to
computational cost reduction of general type-2 fuzzy systems. Engineering Applications of Artificial Intelligence, 2018, 74: 186-197.

[28] M. A. Khanesar, A. Jalalian, O. Kaynak. Improving the speed of center of set type-reduction in interval type-2 fuzzy systems by eliminating the need for sorting. IEEE Transactions on Fuzzy Systems, 2017, 25(5): 1193-1206.

[29] Y. Chen, J. X. Yang. Study on center-of-sets type-reduction of interval type-2 fuzzy logic systems with noniterative algorithms. Journal of Intelligent & Fuzzy Systems, Accept, 2021.

[30] E. Ontiveros, P. Melin, O. Castillo. Comparative study of interval type-2 and general type-2 fuzzy systems in medical diagnosis. Information Sciences, 2020, 525: 37-53.

[31] T. Wang, Y. Chen, S. C. Tong. Fuzzy reasoning models and algorithms on type-2 fuzzy sets. International Journal of Innovative Computing, Information and Control, 2008, 4(10): 2451-2460.

[32] Y. Chen. Study on sampling-based discrete noniterative algorithms for centroid type-reduction of interval type-2 fuzzy logic systems. Soft Computing, 2020, 24(15): 11819-11828.

[33] Y. Chen. Study on sampling based discrete Nie-Tan algorithms for computing the centroids of general type-2 fuzzy sets. IEEE Access, 2019, 7(1): 156984-156992.

[34] Y. Chen. Study on weighted-based discrete noniterative algorithms for computing the centroids of general type-2 fuzzy sets. International Journal of Fuzzy Systems, 2021, accept.

[35] A. Khosravi, S. Nahavandi. Load forecasting using interval type-2 fuzzy logic systems: optimal type reduction. IEEE Transactions on Industrial Informatics, 2014, 10(2): 1055-1063.

[36] T. Zhao, J. Xiao. State feedback control of interval type-2 Takagi-Sugeno fuzzy systems via
interval type-2 regional switching fuzzy controllers. International Journal of Systems Science, 2015, 46(15): 2756-2769.

[37] F. Gaxiola, P. Melin, F. Valdez, J. R. Castro, O. Castillo. Optimization of type-2 fuzzy weights in backpropagation learning for neural networks using GAs and PSO. Applied Soft Computing, 2016, 38: 860-871.

[38] O. Castillo, P. Melin, E. Ontiveros, et al. A high-speed interval type 2 fuzzy system approach for dynamic parameter adaptation in metaheuristics. Engineering Applications of Artificial Intelligence, 2019, 85: 666-680.

[39] C. H. Hsu, C. F. Juang. Evolutionary robot wall-following control using type-2 fuzzy controller with species-de-activated continuous ACO. IEEE Transactions on Fuzzy Systems, 2013, 21(1): 100-112.

[40] H. Mo, F. Y. Wang, M. Zhou, et al. Footprint of uncertainty for type-2 fuzzy sets. Information Sciences, 2014, 272, 96-110.

[41] F. Y. Wang, H. Mo. Some fundamental issues on type-2 fuzzy sets. Acta Automatica Sinica, 2017, 43(7): 1114-1141.

[42] S. C. Tong, Y. M. Li. Observer-based adaptive fuzzy backstepping control of uncertain pure-feedback systems. Science China Information Sciences, 2014, 57(1): 1-14.

[43] Y. Chen, J. X. Yang. Design of back propagation optimized Nagar-Bardini structure based interval type-2 fuzzy logic systems for fuzzy identification. Transactions of the Institute of Measurement and Control, 2021, 43(12): 2780-2787.

[44] E. Ontiveros-Robles, P. Melin, O. Castillo. Comparative analysis of noise robustness of type 2 fuzzy logic controllers. Kybernetika, 2018, 54(1): 175-201.
[45] H. Mo, F. Y. Wang, M. Zhou, et al. Footprint of uncertainty for type-2 fuzzy sets. Information Sciences, 2014, 272, 96-110.