Closed-Form Mathematical Representations of Interval Type-2 Fuzzy Logic Systems with Application to Inverted Pendulum Stabilization

Sherif M. Abuelenin and Rabab F. Abdel-Kader

Port-Said University, Port-Fouad, Port-Said, Egypt

ABSTRACT
Interval type-2 fuzzy logic systems (IT2 FLSs) have a wide range of applications due to their ability to handle uncertainties. A drawback of IT2 FLSs is that their different representations generally suffer from high computational complexity. This paper discusses the representation of IT2 FLSs in closed mathematical form. The proposed form provides an easy, computationally efficient method to implement IT2 FLSs. Additionally, the closed-form is preferred in control systems. We also discuss the use of the proposed method in balancing an inverted pendulum on a cart, a common problem in control theory. Simulation results are provided to compare the performance of the proposed closed-form systems against the performance of other IT2 FLSs. The results confirm that the introduced IT2 representations perform closely to their more computationally complex counterparts.

1. Introduction
Type-2 fuzzy sets (T2 FSs) were proposed by Zadeh in 1975 [1] as an extension of type-1 fuzzy sets (T1 FSs) to better handle uncertainties. The main difference between a T2 FS and a T1 FS is that the membership grades are not crisp numbers but are also fuzzy sets (secondary membership functions) [2–4] making the T2 FS a three-dimensional set. The footprint of uncertainty (FOU) of a T2 FS represents the uncertainties in defining its position and shape. When the secondary membership function (MF) is a unit interval for all the points in the primary membership, the set is called an interval type-2 fuzzy set (IT2 FS) [5,6]. An IT2 FS is completely defined by its FOU because no information is contained in the third dimension [5,7].

A fuzzy system that utilizes IT2 FSs in the antecedent or consequent part of its rule base is known as an interval type-2 fuzzy logic system (IT2 FLS) [3,8]. IT2 FLSs have been used in several applications [6,8–15,45–47], and they often outperform type-1 (T1) systems when handling uncertainties. The output of the inference step of an IT2 FLS is an IT2 FS, which
requires conversion to a T1 FS before the final crisp output can be calculated. This conversion process is called type-reduction (TR) \cite{16,17}. Originally, TR was performed using computationally intensive Karnik-Mendel (KM) iterative algorithms \cite{18}. Several methods were proposed to reduce the computational cost of KM algorithms (see, for example, \cite{19–22} and references therein). Another approach was to replace the KM algorithm with non-iterative alternative architectures. Both categories of methods have demonstrated good performance, and the question of which method (or category of methods) is better, largely depends on the application \cite{23}. The problems of reducing the computational cost and simplifying the implementation of T2 FLSs are still pertinent \cite{24,25}.

In this paper, we propose to use closed-form mathematical representations of IT2-FLSs to provide simplified implementation of such systems with reduced computational cost. Accordingly, we introduce two alternative expressions, each of them fully represents an IT2-FLS. Besides the implementation simplicity and the low-computational cost of the proposed inference mechanisms, their closed-form nature makes them good candidates in control applications where stability analysis is required. The two proposed representations are based on approximating two existing noniterative methods, namely, Coupland and John’s geometric centroid (GC) method \cite{30–36} and Nie-Tan (NT) method \cite{37}. In the rest of this article, we introduce an approach to formulate a closed-form mathematical representation for IT2 FLSs based on the GC method. Then, we extend the results to provide another expression based on the NT operator.

The remainder of this paper is organized as follows. In section 2, we provide background and related work information on IT2 FLSs with a detailed review of the GC and NT methods and the involved mathematical formulations. Section 3 introduces the closed-form representations. Sections 4 and 5 discuss the performance of the introduced systems, followed by conclusions in section 6. A summary of the symbols and the abbreviations used in the paper and their meanings is provided in the appendix.

## 2. Background and Related Work

As mentioned above, the Karnik-Mendel (KM) iterative algorithm is computationally intensive, therefore, several attempts were proposed to reduce its complexity \cite{21,48}. For example, the well-known enhanced KM (EKM) algorithm reduced the number of needed iterations \cite{50}. The iterative algorithm with stopping condition (IASC), and its enhanced version (EIASC) were proposed to further reduce the computational complexity \cite{21}. The EIASC method was reported to be superior to the EKM algorithm but heavily relied on loops \cite{48}.

Alternatively, other works proposed to replace the KM algorithm with noniterative architectures. A summary of many alternative approaches to TR and a comparison of their computational cost were discussed in \cite{21,23,48}. John and Coupland’s Geometric-Centroid method was one of the first methods to avoid the iterative type-reduction algorithms \cite{30–36}. Nie-Tan method later provided a much more accurate IT2 defuzzification method \cite{41}. The GC defuzzification provides an approximation to the TR step \cite{38} by finding the center of area (CoA) of the FOU of the output IT2 FS. By contrast, the NT method computes the average T1 FS of the upper and lower bounds of the FOU.

Having an IT2 closed-form fuzzy system can lead to reduced computational complexity. Also, such system is preferred in control design, especially when stability analysis is required \cite{26–28}. The iterative KM algorithms cannot provide such a closed-form for type-2
systems; therefore, alternative approaches must be utilized [29]. And generally, alternative methods do not directly provide a simple closed mathematical form that fully represents an IT2 FLS. An example of this is discussed in [29], where a closed-form inference mechanism was developed for an IT2 Takagi–Sugeno–Kang fuzzy logic controller based on the uncertainty bounds approximation. In this article, we discuss in detail using the GC and the NT approaches to develop closed-form representations of IT2-FLSs.

For a two-input single-output IT2 FLS. The domain of each input is partitioned into \( W \) IT2 FSs. A total of \( M = W^2 \) possible rules are included in the rule base. The \( k^{th} \) rule (denoted \( R^k \)) is given by

\[
R^k : \text{IF } x_1 \text{ is } \tilde{F}_1^i \text{ AND } x_2 \text{ is } \tilde{F}_2^j \text{ THEN } y \text{ is } \tilde{G}^{ij} \quad (i, j = 1, 2, \ldots, W)
\]

where \( k = (i - 1)W + j, \) and \( k \in [1, M], \) [3]. The \( k^{th} \) rule can also be referred to as the \( ij^{th} \) rule (i.e. knowing \( k \) we implicitly know the corresponding values of \( i \) and \( j \)). The consequents \( \tilde{G}^{ij} \) are, in general, IT2 FSs but can also be T1 fuzzy sets, simple intervals, or numbers. For singleton fuzzification, the firing set (firing level) of \( R^k \) is given by (2).

\[
F^{ij}(x) = T(\mu_{\tilde{F}_1^i}(x_1), \mu_{\tilde{F}_2^j}(x_2)) = \mu_{\tilde{F}_1^i}(x_1) \star \mu_{\tilde{F}_2^j}(x_2)
\]

where \( T(.) \) and ‘\( \star \)’ indicate the utilized t-norm operator (commonly, the minimum or product operators are used). It follows that

\[
F^{ij}(x) = [\mu_{\tilde{F}_1^i}(x_1) \star \mu_{\tilde{F}_2^j}(x_2), \tilde{\mu}_{\tilde{F}_1^i}(x_1) \star \tilde{\mu}_{\tilde{F}_2^j}(x_2)] \equiv [\tilde{f}^{ij}, \tilde{f}^{ij}]
\]

where \( \tilde{f}^{ij} \) and \( \tilde{f}^{ij} \) are, respectively, the lower and upper firing degrees of the \( k^{th} \) rule, and \( \mu_{\tilde{F}_1^i}(x) \) and \( \tilde{\mu}_{\tilde{F}_1^i}(x) \) are the lower and upper membership grades of \( \tilde{F}_1(x) \). When the consequent of \( R^k \) is a T1 FS, \( \tilde{G}^k \) [30], the implied fuzzy set \( \mu_{\tilde{G}^k}(y) \) is given by (4). Up to this step, the upper and lower bounds, \( \mu_{\tilde{G}^k}(y), \tilde{\mu}_{\tilde{G}^k}(y) \) (alternatively called the upper and lower output MFs (UMF and LMF)) have been reached independently using type-1 operations.

\[
\mu_{\tilde{G}^k}(y) = [\mu_{\tilde{G}^k}(y), \tilde{\mu}_{\tilde{G}^k}(y)] = [\tilde{f}^{ij} \star \mu_{\tilde{G}^k}(y), \tilde{f}^{ij} \star \tilde{\mu}_{\tilde{G}^k}(y)]
\]

The next step is to find the overall output IT2 FS by aggregating the implied sets from all the fired rules. Assuming that all the rules are being fired (the non-fired rules can be considered to be fired with zero firing strengths), the output fuzzy set is given by (5). Here, we use ‘\( \sqcup \)’ to represent the utilized aggregation operator. Since we use type-1 operations to find each of the two bounds, it is natural to utilize any of the type-1 aggregation operators (maximum, sum, or probabilistic or are commonly used with T1 FSs [39]).

\[
\mu_{\tilde{G}^k}(y) = \sqcup_{k=1}^M \mu_{\tilde{G}^k}(y)[\sqcap_{k=1}^M (\tilde{f}^{ij} \star \mu_{\tilde{G}^k}(y)), \sqcap_{k=1}^M (\tilde{f}^{ij} \star \tilde{\mu}_{\tilde{G}^k}(y))]
\]

### 2.1. The Geometric Centroid Method

In the GC method, TR is bypassed by finding the x-coordinate of the center of area (CoA) of the FOU of the output IT2 FS (the area bounded by \( \mu_{\tilde{G}^k}(y) \) and \( \tilde{\mu}_{\tilde{G}^k}(y) \)), an extension approach to using the CoA in the defuzzification of T1 FSs [34]. The two bounds UMF(\( \tilde{G} \)) and LMF(\( \tilde{G} \))
are given by (6) and (7), respectively.

$$\mu_{\tilde{B}}(y) = \bigcup_{k=1}^{M} (\tilde{F}_k \star \mu_{G_k}(y))$$

(6)

$$\mu_{\tilde{B}}(y) = \bigcup_{k=1}^{M} (\tilde{F}_k \star \mu_{G_k}(y)) = \bigcup_{k=1}^{M} (\tilde{F}_k \star \mu_{G_k}(y))$$

(7)

Mathematically, the area of the FOU is found by integrating \( dA \) with respect to the output variable \( y \) (see Figure 1). Hence, the CoA can be expressed [40] as the weighted average operation shown in (9).

$$dA = (\tilde{\mu}_{\tilde{B}}(y) - \mu_{\tilde{B}}(y)) \cdot dy$$

(8)

$$y_{Crisp} = \text{CoA}(\tilde{B}) = \frac{\int_{-\infty}^{\infty} y \cdot dA}{\int_{-\infty}^{\infty} dA} = \frac{\int_{-\infty}^{\infty} y \cdot (\tilde{\mu}_{\tilde{B}}(y) - \mu_{\tilde{B}}(y)) \cdot dy}{\int_{-\infty}^{\infty} (\tilde{\mu}_{\tilde{B}}(y) - \mu_{\tilde{B}}(y)) \cdot dy}$$

(9)

When the output MFs of the fuzzy systems are T1 sets, Eq. (9) can be simplified to

$$\frac{\int_{-\infty}^{\infty} y \cdot (\tilde{\mu}_{\tilde{B}}(y)) \cdot dy - \int_{-\infty}^{\infty} y \cdot (\mu_{\tilde{B}}(y)) \cdot dy}{\int_{-\infty}^{\infty} (\tilde{\mu}_{\tilde{B}}(y) - \mu_{\tilde{B}}(y)) \cdot dy} = \frac{C_u A_u - C_l A_l}{A_u - A_l}$$

(10)

where \( C_u, C_l, A_u \) and \( A_l \) represent the centers and the areas of the upper and lower MFs of \( \tilde{B}(y) \) [26–28].

The integration process involved in (9) is computationally intensive. The integration process can be replaced by a discretized weighted average operation (discretized in \( n \) points) to reduce its complexity, as is done for T1 FLSs [30]. In this case, Eq. (9) can be approximated as follows:

$$\text{CoA}(\tilde{B}) \approx \frac{\sum_{i=1}^{n} y_i \mu_{\tilde{B}}(y_i)}{\sum_{i=1}^{n} \mu_{\tilde{B}}(y_i)}$$

(11)

Notably, in John and Coupland’s original GC method [30,35], the FOU is first represented using regular polygons, mostly triangles; then, the weighted average of the polygons
is calculated to replace the integration process. This process is followed to circumvent
the approximation errors associated with numerical integration. Our proposed method
is understood to be an approximate method. In the following section, we discuss the
approximations involved in reaching the required representation of the IT2 FS, including
approximations of the integration process.

2.2. The Nie-Tan Method

The NT operator [37] is another alternative method to TR. It was recently shown [41] that the
NT method provides an accurate IT2 defuzzification. The method outputs a T1 FS that is the
average of the UMF and LMF of the FOU of the output IT2 FS \( \tilde{A} \) whose membership function
is given by \( \mu^*(y) = \frac{1}{2}(\tilde{\mu}(y) + \check{\mu}(y)) \). The final system output is then found by computing
the center of gravity of \( \tilde{A} \).

3. Closed-Form Mathematical Representation of IT2 FLSs

In this section we discuss the derivation of the two IT2 FLS closed-form representations. We
begin by deriving the GC-based approximation, then we extend the results to derive the
NT-based one.

3.1. Derivation of the GC-based Expressions

Equation (11) provides an expression that represents the output of an IT2 FLS using the
CoA of the output FOU as an alternative IT2 defuzzification method. As discussed in the
background section, the arithmetical steps involved in computing both the upper and lower
MFs are mostly independent of each other. Type-1 operations are utilized in reaching both
the UMF and LMF. By following the same reasoning, we can use other T1 techniques to
further simplify the representation of the IT2 FLS.

When using center-average defuzzification, the shapes of the MFs for the output fuzzy
sets have no effect on the final result; therefore, one can use singletons centered at the
appropriate positions [42]. In (8) and (9), the sum operator is used for join, product for t-
norm, and utilize singleton fuzzy MFs \( \mu_{Gk}(y) = \delta(y - b_k) \) for the outputs \( G_k \), where ‘\( \delta \)’ is
the Dirac delta function used to represent the fuzzy singleton, and \( b_k \) are the locations of the
output MFs (this is equivalent to using center-average defuzzification with \( b_k \) representing
the centers of the output MFs). Substituting into (11), the output is rewritten as

\[
y_{crispCG} = \frac{\sum_{k=1}^{M} b_k \left( \tilde{\mu}_{\tilde{F}_1}(x_1) \cdot \check{\mu}_{\check{F}_2}(x_2) \right) - \sum_{k=1}^{M} b_k \left( \mu_{\tilde{F}_1}(x_1) \cdot \check{\mu}_{\check{F}_2}(x_2) \right)}{\sum_{k=1}^{M} \tilde{\mu}_{\tilde{F}_1}(x_1) \cdot \check{\mu}_{\check{F}_2}(x_2) - \sum_{k=1}^{M} \mu_{\tilde{F}_1}(x_1) \cdot \check{\mu}_{\check{F}_2}(x_2)}
\]

(Note that we use ‘\( k \)’ in the notation \( b_k \) simply as a label for each rule (i.e. we number the rules
in the rule-base, and \( k \) is this number). Hence, when we are given \( k \), we know the values of
the corresponding \( i \) and \( j \).

Equation (12) can be represented in a more compact form:

\[
y_{crispCG} = \frac{\sum_{k=1}^{M} b_k \left( \tilde{\mu}_{\tilde{F}_1}(x_1) \cdot \check{\mu}_{\check{F}_2}(x_2) - \mu_{\tilde{F}_1}(x_1) \cdot \check{\mu}_{\check{F}_2}(x_2) \right)}{\sum_{k=1}^{M} \tilde{\mu}_{\tilde{F}_1}(x_1) \cdot \check{\mu}_{\check{F}_2}(x_2) - \sum_{k=1}^{M} \mu_{\tilde{F}_1}(x_1) \cdot \check{\mu}_{\check{F}_2}(x_2)}
\]  

(13)
Carefulness is required when designing the fuzzy system so that the denominator of (13) is not equal to zero for any input value. Additionally, if different output singletons are used for the upper and lower bounds, the crisp output of (12) can be rewritten as in (14). This process represents a whole IT2 FLS in a simple closed-form formulation, which can be considered an extension to the T1 closed-form counterparts (as discussed in [42]).

\[
y_{\text{crispGC}} = \frac{\sum_{k=1}^{M} \tilde{b}_k \left( \tilde{\mu}_{\tilde{F}_1}(x_1) \cdot \tilde{\mu}_{\tilde{F}_2}(x_2) \right) - \sum_{k=1}^{M} \tilde{b}_k (\mu_{\tilde{F}_1}(x_1) \cdot \mu_{\tilde{F}_2}(x_2))}{\sum_{k=1}^{M} (\tilde{\mu}_{\tilde{F}_1}(x_1) \cdot \tilde{\mu}_{\tilde{F}_2}(x_2) - \mu_{\tilde{F}_1}(x_1) \cdot \mu_{\tilde{F}_2}(x_2))} \quad (14)
\]

### 3.2. NT-Based Expression

An alternative closed-form mathematical expression can be obtained by utilizing the NT operator. As mentioned above, the NT operator outputs a T1 FS that is the average of the UMF and LMF of the FOU of the output IT2 FS \( \tilde{A} \), as given by Eq. (15). The final system output is then found by computing the center of gravity of \( \tilde{A} \) using (16).

\[
\mu^s(y) = \frac{1}{2} (\tilde{\mu}(y) + \underline{\mu}(y)) \quad (15)
\]

\[
C(\tilde{A}) = \frac{\int_{y_{\min}}^{y_{\max}} \mu^s(y) \cdot y \, dy}{\int_{y_{\min}}^{y_{\max}} \mu^s(y) \, dy} \quad (16)
\]

Using (15) and following a similar approach to our earlier discussion, another closed-form approximation for the IT2 FLS, given in (17), that is based on the NT method can be reached.

\[
y_{\text{crispNT}} = \frac{\sum_{k=1}^{M} \tilde{b}_k \left( \tilde{\mu}_{\tilde{F}_1}(x_1) \cdot \tilde{\mu}_{\tilde{F}_2}(x_2) \right) + \sum_{k=1}^{M} \tilde{b}_k (\mu_{\tilde{F}_1}(x_1) \cdot \mu_{\tilde{F}_2}(x_2))}{\sum_{k=1}^{M} (\tilde{\mu}_{\tilde{F}_1}(x_1) \cdot \tilde{\mu}_{\tilde{F}_2}(x_2) + \mu_{\tilde{F}_1}(x_1) \cdot \mu_{\tilde{F}_2}(x_2))} \quad (17)
\]

If different output singletons are used for the upper and lower bounds, the crisp output of (17) can be rewritten as in (18).

\[
y_{\text{crispNT}} = \frac{\sum_{k=1}^{M} \tilde{b}_k \left( \tilde{\mu}_{\tilde{F}_1}(x_1) \cdot \tilde{\mu}_{\tilde{F}_2}(x_2) \right)}{\sum_{k=1}^{M} (\tilde{\mu}_{\tilde{F}_1}(x_1) \cdot \tilde{\mu}_{\tilde{F}_2}(x_2) + \mu_{\tilde{F}_1}(x_1) \cdot \mu_{\tilde{F}_2}(x_2))} \quad (18)
\]

This can be generalized (for both CG and NT) to multiple-input systems as follows:

\[
y_{\text{crisp}} = \frac{\sum_{k=1}^{M} \left( \tilde{b}_k \cdot \prod_{n=1}^{N} \tilde{\mu}_{\tilde{F}_n}(x_n) \right) + \sum_{k=1}^{M} \left( b_k \cdot \prod_{n=1}^{N} \mu_{\tilde{F}_n}(x_n) \right)}{\sum_{k=1}^{M} \left( \prod_{n=1}^{N} \tilde{\mu}_{\tilde{F}_n}(x_n) \pm \prod_{n=1}^{N} \mu_{\tilde{F}_n}(x_n) \right)} \quad (19)
\]

where \( N \) is the total number of inputs in the system.

### 3.3. Selection of Membership Functions

Both Eqs. (13) and (17) represent a whole IT2 FLS in an explicit mathematical expression. Eq. (13) gives an approximate closed-form GC based (CFGC) IT2 system, and eq. (17) gives
an approximate closed-form NT method (CFNT) based system. In principle, any MF may be used in these expressions. However, piecewise MFs (e.g. triangular and trapezoidal) should be avoided to obtain a ‘true’ and simple mathematical form. The use of such MFs would complicate the expressions because defining these functions mathematically would require the inclusion of operators such as max and min or the utilization of discontinuous functions such as the Heaviside unit step function. Continuous MFs, such as the Gaussian, are used instead. Nevertheless, in the results section, we study the performance of the proposed systems with triangular MFs as well as with Gaussian MFs.

As stated above, any form of MF can be used with the two developed fuzzy mechanisms i.e. CFGC and CFNT, as represented by Eqs. (13) and (17), respectively. However, to ensure a true and simple mathematical representation (i.e. to avoid the complexity of representing discontinuities associated with expressing MFs such as trapezoidal and triangular), we limit the MFs to be Gaussian.

\[
\mu_F(x) = \exp \left( -\frac{1}{2} \left( \frac{x - m_i}{\sigma_i} \right)^2 \right)
\]  

(20)

The IT2 FSs used can be uncertain in means (centers) or uncertain in variances (widths). The UMFs and LMFs are simpler to represent when the utilized IT2 MFs have certain means and uncertain variances [43]. For a given IT2 set, both the UMF and LMF are Gaussian with the same mean and different variances. By contrast, for an IT2 set with uncertain mean, neither the UMF nor the LMF is Gaussian (see Figure 2 for illustration). Therefore, to maintain the mathematical simplicity of the IT2 FLS expression, the UMFs and LMFs must be redefined to avoid using max and min operators. As shown in Figure 2, curve fitting techniques can be used to approximate both the UMF and LMF with Gaussian equivalents. In the figure, the LMF is exactly described by (21), while the approximate LMF is described by (22). Similar expressions can be written for the UMF.

\[
\bar{\mu}(x) = \min \left( \exp \left( -\frac{1}{2} \left( \frac{x + \Delta \mu}{\sigma} \right)^2 \right), \exp \left( -\frac{1}{2} \left( \frac{x - \Delta \mu}{\sigma} \right)^2 \right) \right)
\]  

(21)

where \( \sigma \) is the fixed standard deviation of a Gaussian membership function with an uncertain mean \( \in [\mu - \Delta \mu, \mu + \Delta \mu] \).

\[
\bar{\mu}(x) \approx \alpha_L \exp \left( -\frac{1}{2} \left( \frac{x}{\sigma_L} \right)^2 \right)
\]  

(22)

where \( \alpha_L \) is the scaling factor for the approximated LMF which can be determined from the FOU, and \( \sigma_L \) is the standard deviation of the approximated LMF, as illustrated in Figure 2.

In summary, Eqs. (13) and (17) are introduced to provide two alternative closed-form simple representations of IT2 FLSs, CFGC and CFNT. The main difference between the two equations is that each is based on a different IT2 FLS architecture. The CFGC, as described by Eq. (13) is based on the ‘Geometric IT2-FSs’ approach, which was initially introduced by Coupland and John in [33]. Their method approximates the crisp output of the system by computing the GC of the FOU of the output IT2 FS. Eq. (13) is a simplified representation of their method. On the other hand, the CFNT, as given by Eq. (17) provides an alternative representation based on the NT approach, which was originally introduced by Nie and Tan.
Figure 2. An example of a Gaussian membership function with uncertain mean \( \in [\mu - \Delta \mu, \mu + \Delta \mu] \) and fixed standard deviation \( \sigma \) (green shaded area, here \( \mu = 0, \Delta \mu = 0.1, \) and \( \sigma = 0.418 \)) and the approximation of its UMF and LMF (black solid lines; the approximate UMF is Gaussian with \( \mu = 0 \) and \( \sigma = 0.4937 \), the LMF is a scaled-down Gaussian with \( \mu = 0, \sigma = 0.3651, \) and a scaling factor of 0.9183).

in [37] and further investigated in [41]. In contrast to the GC method, the NT method computes a T1 FS that is an average of the UMF and LMF of the output IT2 FS. The crisp output is computed as the center of gravity of this T1 set. Therefore, Eq. (17) is a simplified representation of the NT method. We also showed how to use approximate Gaussian functions for both the UMF and LMF to simplify the representation when the utilized IT2 MFs have uncertain means.

4. Performance Evaluation of the Closed-form Systems

In this section, we evaluate the performance of the proposed closed-form system by studying the control surfaces resulting from the proposed systems and their corresponding computational complexity. Then an application Example ‘Control of an Inverted Pendulum’ is discussed to show how the proposed closed-form mechanisms can be used to design an IT2 fuzzy logic controller (FLC) in a common control problem.

All the results introduced in this section are based on MATLAB® and Simulink® simulations. The simulations were performed in MATLAB 2017a using a laptop PC with an Intel core i3-7100 CPU @2.40 GHz and 6 GB RAM running Windows 10 Home 64- bit. MATLAB® implementation of the proposed methods is available online in [49].

4.1. Computational Complexity

There are different approaches of measuring the computational complexity of an algorithm, the common ones are (i) counting the arithmetic operations (or alternatively, the number of the floating-point operations) in the algorithm, and (ii) calculating the execution time it takes to run the algorithm on a specific machine [21,23]. The first method is suitable if all algorithms to be compared are similar in form (e.g. iterative, recursive, closed-form, etc.). The second method is more suitable when the different algorithms to be studied are different. As mentioned above, to the best of our knowledge, there are very few
Table 1. Rule base for three membership functions.

| Rule number | 1st input | 2nd input | Output | Corresponding $b_i$ value |
|-------------|-----------|-----------|--------|--------------------------|
| 1           | N         | N         | P      | 1                        |
| 2           | N         | Z         | P      | 1                        |
| 3           | N         | P         | Z      | 0                        |
| 4           | Z         | N         | P      | 1                        |
| 5           | Z         | Z         | Z      | 0                        |
| 6           | Z         | P         | N      | −1                       |
| 7           | P         | N         | Z      | 0                        |
| 8           | P         | Z         | N      | −1                       |
| 9           | P         | P         | N      | −1                       |

Table 2. Performance comparison of the proposed methods and the EIASC algorithm.

|                              | Average run time for computing the control surface (in seconds) | Mean of the absolute difference of the control surface | Standard deviation of the absolute difference of the control surface |
|------------------------------|------------------------------------------------------------------|-------------------------------------------------------|---------------------------------------------------------------------|
| The EIASC algorithm (Reference) with triangular input membership functions | 0.2668                                                            | 0                                                      | 0                                                                  |
| The Proposed CFGC algorithm (Eq. (13)) with triangular input membership functions | 0.1093                                                            | 0.1223                                                | 0.1004                                                             |
| Proposed CFNT algorithm (Eq. (17)) with triangular input membership functions | 0.1086                                                            | 0.0183                                                | 0.0102                                                             |
| Proposed CFGC algorithm (Eq. (13)) with Gaussian input membership functions | 0.0190                                                            | 0.1143                                                | 0.1050                                                             |
| Proposed CFNT algorithm (Eq. (17)) with Gaussian input membership functions | 0.0194                                                            | 0.0275                                                | 0.0251                                                             |

closed-form IT2 systems that were previously discussed in literature, specifically, those discussed in [29], which are very similar in the number of operations to the two closed-forms introduced in this article. The main difference is that the work presented in [29] was on Takagi–Sugeno–Kang fuzzy logic controller based on the uncertainty-bounds approximation. Therefore, we rely on execution time for complexity comparison with other algorithms.

First, we compare the proposed closed-form systems with an enhanced KM algorithm. Two aspects are considered: the control surface and the execution time. We selected the enhanced KM iterative algorithm with stop condition (EIASC) [21] because it was recommended for use in practical applications due to its ease of implementation. EIASC was shown [21] to be significantly faster than the original KM and other enhanced KM algorithms, especially for small rule bases.

Table 2 compares the EIASC algorithm and the CFGC expression (Eq. (13)) and the CFNT expression (Eq. (17)) in terms of the computation time and the absolute difference statistics of the generated control surfaces. The computational time, in seconds, is the average of 10 runs of each algorithm (after excluding outliers). Each run computes a 441-point control surface ($21 \times 21$ points) of a two-input single-output system. The difference statistics provide the mean and the standard deviation of the absolute difference between the control surface of the controller and that of the reference (EIASC) controller. The table shows that the
The three membership functions for one of the inputs $x_1$. By limiting the range of the input to values $\in [-1, 1]$, the left and right membership functions are practically $S$ and $Z$ Gaussian. The three IT2 membership functions are Gaussian with uncertain means. The means are $\mu_1 = -1$, $\mu_2 = 0$, and $\mu_3 = 1$, the uncertainty around the means is $\Delta \mu = 1/8$, and the standard deviation is $\sigma = 0.418$. (Note that the approximate UMFs are Gaussian with $\sigma_U = 0.5128$, and the approximate LMF is a scaled-down Gaussian, with $\sigma_L = 0.3532$ and a scaling factor of $\sigma_L = 0.895$.)

proposed forms have substantially lower computation times, particularly when Gaussian MFs are used.

### 4.2. Control Surface

Equations (13) and (17) were used to represent a simple two-input single-output IT2 fuzzy system. The system utilizes three MFs in the domain of each input. Figure 3 illustrates the three MFs utilized in one of the input domains (both are identical). The rule base is shown in table 1, with the corresponding $b_i$ values. The three MFs have uncertain means that vary between $\mu_i - \Delta \mu$ and $\mu_i + \Delta \mu$ ($\mu_1 = -1$, $\mu_2 = 0$, $\mu_3 = 1$, $\Delta \mu = 1/8$, and $\sigma = 0.418$). Each UMF and LMF was approximated using Gaussian functions, as described in the previous section, to use Eq.(13) and Eq.(17). The approximate UMFs are as follows (with the same values of $\mu_i$) with $\sigma_U = 0.5128$. Each LMF is a scaled-down Gaussian with $\sigma_L = 0.3532$ and an amplitude scaling factor of $\sigma_L = 0.895$.

Figures 4–7 show the resulting surfaces of the introduced systems of (13) and (17) with both the exact and approximate forms of the UMFs and LMFs. Input and output gains may be introduced to scale the axes of the control surface as needed. In addition to the results given in table 2, the figures provide insight into the resemblances of the control surfaces of the approximated proposed systems to their counterparts.

### 4.3. Application Example: Control of an Inverted Pendulum

In this section, we discuss how the proposed closed-form mechanisms can be used to design an IT2 fuzzy logic controller (FLC) in a common control problem.
Figure 4. Surface using the introduced CFGC representation (Eq. (13)) with the approximated Gaussian functions.

Figure 5. Surface using the introduced CFGC representation (Eq. (13)) with the exact representations of the UMF and LMF.
Figure 6. Surface using the introduced CFTN representation for (Eq. (17)) with the approximated Gaussian functions.

Figure 7. Surface using the introduced CFTN representation (Eq. (17)) with the exact representations of UMF and LMF.
Balancing an inverted pendulum is considered a benchmarking problem in control [44]. We utilize the two introduced equations (i.e. (13) and (17)) in a feedback control scheme to balance an inverted pendulum and compare the performance to that of the two reference IT2 methods (i.e. NT and GC). The objective of the control problem is to balance a pendulum with an initial nonzero angle in the upright position (i.e. $r = 0$).

An inverted pendulum mounted on a cart is balanced by applying dynamic force $f$ to the cart. The error signal (the angular difference between the upright position and the actual position of the pendulum) and its derivative are commonly used as inputs to the feedback controller utilized in balancing the pendulum. Figure 9(a,b) show the block diagram of the control system and the system model implemented using Simulink respectively. As mentioned earlier, gains are introduced to scale the universes of discourse for the fuzzy MFs. Here, the values of $g_1$, $g_2$, and $g_y$ are $4/\pi$, $0.4/\pi$, and 100, respectively. We use the model described by (23) and (24) for the inverted pendulum system [42].

\[
\ddot{y} = \left( g \sin(y) + \cos(y) \left[ -\dot{f} - 0.25y^2 \sin(y) \right] \right) / \left( \frac{2}{3} - \frac{1}{6} \cos^2(y) \right)
\]

(23)

\[
\dot{f} = -100\ddot{f} + 100f
\]

where $f$ is the applied force in newtons, $y$ is the angular position in radians, and $g$ is the gravity of Earth.

5. Results and Discussion

The main objective of the proposed closed-form representation of the interval type-2 fuzzy systems was to provide a more compact, easy, and computationally efficient method to
implement IT2 FLS as compared to the traditional representation. The proposed closed-form representation was compared to EIASC algorithm using two performance evaluation methods as explained in the previous section computation complexity, and control surfaces comparison. The EIASC algorithm was chosen since it was reported to be one of the superior algorithms in terms of computation complexity [48] compared to other iterative TR methods.

First, the computation complexity is calculated for the proposed method and other competing algorithms by measuring the execution time it takes to run the algorithm on a specific machine. Table 2 lists the computation complexity of the proposed methods and EIASC. It is noticed that the computation complexity is reduced by at least 60% when the proposed CFGC algorithm (Eq. (13)) with triangular input membership functions is used and a maximum computation complexity reduction of almost 93% when the proposed CFGC algorithm (Eq. (13)) with Gaussian input membership functions is used as compared to EIASC. This intensive computation complexity reduction can be explained by the absence of loops or iterations in the proposed method compared to EIASC.

Next, a comparison of the generated control surfaces. Figures 4 through 8 show the control surfaces using the introduced CFGC (Eq.(13)) with the approximated Gaussian functions, the control surfaces for the introduced CFGC (Eq.(13)) with the exact representations of the UMF and LMF, the control surface using the introduced CFNT representation (Eq.(17)) with approximated Gaussian functions, the control surface using the CFNT representation (Eq.(17)) with the exact representations of UMF and LMF, and finally the control surfaces using the EIASC algorithm. Table 2 illustrates the values for the mean of the absolute difference of the control surfaces, and the standard deviation of the absolute difference of the control surfaces as a quantitative method to measure the resemblance. The Table shows that the control surfaces generated by the proposed CFNT for either triangular or Gaussian input membership functions are closer to the EIASC control surface as compared to the CFGC for the same type of input membership functions. This agrees with what was reported in literature that the NT approach provides a more accurate TR alternative [41].

Figure 9. (a) Block-diagram model of inverted pendulum control. (b) System model implemented using Simulink
summarize the findings, the simulations proved that the proposed closed-form representation significantly reduces the computation complexity for a small difference in the control surfaces.

Figure 10 shows the MATLAB simulation results of controlling this inverted pendulum utilizing the GC method (numerically evaluated) and utilizing the two different realizations of CFGC discussed earlier (i.e. using exact Gaussian MFs and using the approximate forms). Figure 11 shows similar results for the NT method and the CFNT. Figures 12 and 13 show the stabilization of the inverted pendulum from an initial nonzero condition with the inclusion of disturbance forces. All figures prove that the introduced formulas perform very closely to their more complex represented counterparts.

Next, the controller performance using CFGC and CFNT (i.e. using exact Gaussian MFs and using the approximate forms) is compared to performance of KM, enhanced KM (EKM), KM iterative algorithm with stop condition (IASC), and enhanced IASC (EIASC) using several performance indices. The mentioned algorithms were implemented using IT2 toolbox provided in MATLAB 2020b. The calculated values for Integral of absolute error (IAE), Integral time absolute error (ITAE), Integrated time square error (ITSE), and Integral of square error (ISE) for the eight mentioned scenarios are listed in Table 3. The results agree with what was proved previously that the introduced formulas perform very closely to their more complex represented counterparts.
Figure 11. Angular position of an inverted pendulum controlled by the NT method and by the proposed CENT representation (Eq. (17)).

Figure 12. Angular position of an inverted pendulum controlled by the GC method and by the proposed CFGC (Eq. (13)) representation under disturbance action.
Figure 13. Angular position of the inverted pendulum controlled by the NT method and by the proposed CFNT (Eq. (17)) representation under disturbance action.

Table 3. Controller performance indices the proposed methods.

| Method                             | IAE      | ITAE     | ITSE     | ISE      |
|------------------------------------|----------|----------|----------|----------|
| GC                                 | 0.0109   | 0.000926 | 0.000031 | 0.000679 |
| NT                                 | 0.0137   | 0.0015   | 0.000054 | 0.000907 |
| KM                                 | 0.0144   | 0.0016   | 0.000054 | 0.000958 |
| EKM                                | 0.0144   | 0.0016   | 0.000054 | 0.000958 |
| IASC                               | 0.0144   | 0.0016   | 0.000054 | 0.000958 |
| EIASC                              | 0.0144   | 0.0016   | 0.000054 | 0.000958 |
| CFG using exact Gaussian MFs       | 0.0109   | 0.000929 | 0.000032 | 0.000684 |
| CFNT using exact Gaussian MFs      | 0.0130   | 0.0012   | 0.000050 | 0.000890 |
| CFG using approximate forms        | 0.0108   | 0.000855 | 0.000033 | 0.000710 |
| CFNT using approximate forms       | 0.0134   | 0.001300 | 0.000052 | 0.000913 |

6. Conclusions

This paper presents explicit closed-form representations for IT2 FLSs that are more compact, easy to implement, and computationally efficient compared to other more complex counterparts. The approach was based on the notion that both the UMF and LMF of the output IT2 FS can be found independently by using the CoA of the FOU as an approximation of the crisp output of an IT2 FLS to obtain a closed-form expression. Then, the same approach was utilized with the NT operator, resulting in a different closed-form mathematical expression.

The manuscript compares the results of the proposed methods (in terms of computational complexity and control performance) to six reference methods, namely Coupland’s Geometric-Centroid method which was one of the first methods to avoid the iterative type-reduction algorithms, Nie-Tan method which provided a much more accurate IT2
defuzzification method, the iterative EIASC method which is considered one of the most efficient iterative type-reduction methods, the original Karnik-Mendel (KM) type-reduction approach, the enhanced-KM (EKM), and the IASC approach. The results proves that the performance of the approximate closed-form representations (both GC and NT) was close to the reference approaches but with two advantages: better computational complexity (because of the non-iterative implementation), and a simpler implementation (single closed-form formulation). In addition, the results shows that the closed-form is also preferable in control systems when stability analysis is required and that the introduced IT2 representations perform closely to their more computationally complex counterparts.

In the future, study could be done to extend the proposed closed-form representation of interval type-2 fuzzy logic systems to general type-2 fuzzy logic systems.

**Disclosure statement**

No potential conflict of interest was reported by the author(s).

**Notes on contributors**

**Sherif M. Abuelenin**, received the B.Sc. degree in electronics and communications from Suez Canal University, Egypt in 1999, the M.Sc. degree in Electrical Engineering from Tuskegee University, AL, USA in 2002, and the Ph.D. degree from Auburn University, AL, USA in 2005. He joined Tuskegee University as an assistant professor of Electrical Engineering in 2005. From 2007 to 2011 he served as an assistant professor in the faculty of Engineering Sciences, Sinai University, Egypt. In 2011, he joined the faculty of engineering, Port-Said University, where he currently is an Associate Professor of Electrical engineering. His research interests include signal processing for communications, computational electromagnetics, vehicular communications, and fuzzy logic systems.

**Rabab Farouk Abdel-Kader** she received her B.S. from the Electrical Engineering Department Suez Canal University in 1998. She received the Ph.D. degree from the department of Computer Science and Software Engineering at Auburn University, Auburn, AL in 2007 and the MS degree in Electrical Engineering from Tuskegee University with high honors in 2002. Since 2008 she is working as an Assistant Professor in the Electrical Engineering department, Faculty of Engineering, Port-Said University, Egypt. Her main research interests include image processing, parallel computing, and software Engineering.

**Funding**

This research did not receive any specific grants from funding agencies in the public, commercial, or not-for-profit sectors.

**ORCID**

Sherif M. Abuelenin [http://orcid.org/0000-0002-6489-7457](http://orcid.org/0000-0002-6489-7457)

Rabab F. Abdel-Kader [http://orcid.org/0000-0002-6719-903X](http://orcid.org/0000-0002-6719-903X)

**References**

[1] Zadeh LA. The concept of a linguistic variable and its application to approximate reasoning. Inf Sci. 1975;8:43–80. doi:10.1016/0020-0255(75)90036-5.

[2] Mendel J. Type-2 fuzzy sets and systems: an overview. IEEE Comput Intell Mag. February 2007;2:20–29. doi:10.1109/MCI.2007.357235.
[3] Wu D, Tan WW. A simplified architecture for type-2 FLSs and its application to nonlinear control. in *Proc. IEEE Conf. On Cybern. and Intell. Syst.*, Singapore, Dec. 2004, pp. 485–490. doi:10.1109/ICCISS.2004.1460463.

[4] Karnik NN, Mendel JM. Operations on type-2 fuzzy sets. Fuzzy Syst. Syst. [Internet]. 2001 [cited 2019 Aug 30]; 122:327–348. Available at: https://www.sciencedirect.com/science/article/pii/S0165011400000798.

[5] Mendel JM, John RIB. Type-2 fuzzy sets made simple. IEEE Trans Fuzzy Syst. Apr. 2002;10(2):117–127. doi:10.1109/91.995115.

[6] Castro JR, Castillo O, Melin P. An Interval Type-2 Fuzzy Logic Toolbox for Control Applications. 2007 IEEE International Fuzzy Systems Conference [Internet]. IEEE; 2007 [cited 2019 Aug 30]. p. 1–6. Available from: http://ieeexplore.ieee.org/document/4295341/.

[7] Mendel JM. Advances in type-2 fuzzy sets and systems. Inf Sci. 2007;177:84–110. doi:10.1016/j.ins.2006.05.003.

[8] Liang Q, Karnik NN, Mendel JM. Connection admission control in ATM networks using survey-based type-2 fuzzy logic systems. IEEE Trans Syst Man Cybern C (Appl Rev). Aug 2000;30(3):329–339. doi:10.1109/5326.885114.

[9] Wu D, Mendel JM. Recommendations on designing practical interval type-2 fuzzy systems. Eng Appl Artif Intell [Internet]. 2019 [cited 2019 Aug 30]; 85:182–193. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0952197619301514.

[10] Huang J, Ri M, Wu D, et al. Interval type-2 fuzzy logic modeling and control of a mobile two-wheeled inverted pendulum. IEEE Trans Fuzzy Syst [Internet]. 2018 [cited 2019 Aug 31]; 26:2030–2038. Available at: https://ieeexplore.ieee.org/document/8060588/.

[11] Hassanzadeh HR, Akbarzadeh-T M-R, Akbarzadeh A, et al. An interval-valued fuzzy controller for complex dynamical systems with application to a 3-PSP parallel robot. Fuzzy Sets Syst [Internet]. 2014 [cited 2019 Aug 30]; 235:83–100. Available at: https://www.sciencedirect.com/science/article/pii/S016501141300081X.

[12] Hagras H, Callaghan V, Colley M, et al. Creating an ambient-intelligence environment using embedded agents. IEEE Intell Syst [Internet]. 2004 [cited 2019 Aug 30]; 19:12–20. Available at: http://ieeexplore.ieee.org/document/1363729/.

[13] Yao B, Hagras H, Lepley JJ, et al. An evolutionary optimization based interval type-2 fuzzy classification system for human behavior recognition and summarisation. 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC) [Internet]. IEEE; 2016 [cited 2019 Aug 30]. p. 004706–004711. Available at: http://ieeexplore.ieee.org/document/7844974/.

[14] Castro JR, Rodríguez-Díaz A. A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks. Inf Sci [Internet]. 2009 [cited 2019 Aug 30]; 179:2175–2193. Available at: https://www.sciencedirect.com/science/article/pii/S002002550800426X.

[15] Acampora G, Alghazzawi D, Hagras H, et al. An interval type-2 fuzzy logic based framework for reputation management in Peer-to-Peer e-commerce. Inf Sci [Internet]. 2016 [cited 2019 Aug 31]; 333:88–107. Available at: https://www.sciencedirect.com/science/article/pii/S0020025515008257.

[16] Karnik NN, Mendel JM, Liang Q. Type-2 fuzzy logic systems. IEEE Trans Fuzzy Syst [Internet]. 1999 [cited 2019 Aug 30]; 7:643–658. Available at: http://ieeexplore.ieee.org/document/811231/.

[17] Mendel JM. *Uncertain rule-based fuzzy systems*. [Internet]. Cham: Springer International Publishing; 2017 [cited 2019 Aug 30]. Available at: http://link.springer.com/10.1007978-3-319-51370-6.

[18] Karnik NN, Mendel JM. Centroid of a type-2 fuzzy set. Inf Sci [Internet]. 2001 [cited 2019 Aug 30]; 132:195–220. Available at: https://linkinghub.elsevier.com/retrieve/pii/S002002551000069X.

[19] Mendel JM, Liu F. Super-exponential convergence of the Karnik–Mendel algorithms for computing the centroid of an interval type-2 fuzzy set. IEEE Trans Fuzzy Syst [Internet]. 2007 [cited 2019 Aug 30]; 15:309–320. Available at: http://ieeexplore.ieee.org/document/4142760/.

[20] Khanesar MA, Khakshour AJ, Kaynak O, et al. Improving the speed of center of sets type reduction in interval type-2 fuzzy systems by eliminating the need for sorting. IEEE Trans Fuzzy Syst [Internet]. 2017 [cited 2019 Aug 30]; 25:1193–1206. Available at: http://ieeexplore.ieee.org/document/7551199/.
[21] Wu D. Approaches for reducing the computational cost of interval type-2 fuzzy logic systems: overview and comparisons. IEEE Trans Fuzzy Syst [Internet]. 2013 [cited 2019 Aug 30]; 21:80–99. Available at: http://ieeexplore.ieee.org/document/6208856/.

[22] Han S, Liu X. Global convergence of Karnik–Mendel algorithms. Fuzzy Sets Syst [Internet]. 2016 [cited 2019 Aug 30]; 283:108–119. Available at: https://www.sciencedirect.com/science/article/pii/S0165011415001281.

[23] Wu D. An overview of alternative type-reduction approaches for reducing the computational cost of interval type-2 fuzzy logic controllers. 2012 IEEE International Conference on Fuzzy Systems [Internet]. IEEE; 2012 [cited 2019 Aug 30]. p. 1–8. Available at: http://ieeexplore.ieee.org/document/6251242/.

[24] Runkler TA, Chen C, John R. Type reduction operators for interval type–2 defuzzification. Inf Sci [Internet]. 2018 [cited 2019 Aug 30]; 467:464–476. Available at: https://www.sciencedirect.com/science/article/pii/S0020025518306285.

[25] Ontiveros E, Melin P, Castillo O. High order $\alpha$-planes integration: A new approach to computational cost reduction of General Type-2 Fuzzy Systems. Eng Appl Artif Intell [Internet]. 2018 [cited 2019 Aug 31]; 74:186–197. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0952197618301441.

[26] Begian MB, Melek WW, Mendel JM. Stability analysis of type-2 fuzzy systems. 2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence) [Internet]. IEEE; 2008 [cited 2019 Aug 30]. p. 947–953. Available at: http://ieeexplore.ieee.org/document/4630483/.

[27] Lam HK, Seneviratne LD. Stability analysis of interval type-2 fuzzy-model-based control systems. IEEE Trans Syst Man Cybern [Internet]. 2008 [cited 2019 Aug 30]; 38:617–628. Available at: http://ieeexplore.ieee.org/document/4522595/.

[28] Castillo O, Aguilar L, Cázarez N, et al. Systematic design of a stable type-2 fuzzy logic controller. Appl Soft Comput [Internet]. 2008 [cited 2019 Aug 30]; 8:1274–1279. Available at: https://www.sciencedirect.com/science/article/pii/S156849460700124X.

[29] Biglarbegian M, Melek WW, Mendel JM. On the stability of interval type-2 TSK fuzzy logic control systems. IEEE Trans Syst Man Cybern [Internet]. 2010 [cited 2019 Aug 30]; 40:798–818. Available at: http://ieeexplore.ieee.org/document/5299179/.

[30] Coupland S, John R. Geometric type-1 and type-2 fuzzy logic systems. IEEE Trans Fuzzy Syst [Internet]. 2007 [cited 2019 Aug 30]; 15:3–15. Available at: http://ieeexplore.ieee.org/document/4088983/.

[31] Coupland S, John R. A new and efficient method for the type-2 meet operation. 2004 IEEE International Conference on Fuzzy Systems (IEEE Cat. No.04CH37542) [Internet]. IEEE; [cited 2019 Aug 30]. p. 959–964. Available at: http://ieeexplore.ieee.org/document/1375537/.

[32] Coupland S, John R. (1955). Fuzzy logic and computational geometry. RASC 2004 [Internet]. Nottingham, U.K.; 2004 [cited 2019 Aug 30]. p. 3–8. Available at: https://dora.dmu.ac.uk/handle/2086/951.

[33] Coupland S, John R. Towards More Efficient Type-2 Fuzzy Logic Systems. The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ ’05. [Internet]. IEEE; [cited 2019 Aug 30]. p. 236–241. Available at: http://ieeexplore.ieee.org/document/1452399/.

[34] Coupland S. Type-2 Fuzzy Sets: Geometric Defuzzification and Type-Reduction. 2007 IEEE Symposium on Foundations of Computational Intelligence [Internet]. IEEE; 2007 [cited 2019 Aug 30]. p. 622–629. Available at: http://ieeexplore.ieee.org/document/4233971/.

[35] Coupland S, John R. A Fast geometric method for defuzzification of type-2 fuzzy sets. IEEE Trans Fuzzy Syst [Internet]. 2008 [cited 2019 Aug 30]; 16:929–941. Available at: http://ieeexplore.ieee.org/document/4505326/.

[36] Coupland S, John R. Geometric Interval Type-2 Fuzzy Systems. EUSFLAT Conf. 2005. p. 449–454.

[37] Nie M, Tan WW. Towards an efficient type-reduction method for interval type-2 fuzzy logic systems. 2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence) [Internet]. IEEE; 2008 [cited 2019 Aug 30]. p. 1425–1432. Available at: http://ieeexplore.ieee.org/document/4630559/.
[38] Yeh C-Y, Jeng W-HR, Lee S-J. An Enhanced type-reduction algorithm for type-2 fuzzy sets. IEEE Trans Fuzzy Syst [Internet]. 2011 [cited 2019 Aug 30]; 19:227–240. Available at: http://ieeexplore.ieee.org/document/5638621/.

[39] Yager RR, Filev DP. Essentials of fuzzy modeling and control. New York. 1994:388.

[40] Weisstein EW. Function centroid [Internet]. Wolfram Research, Inc.; [cited 2019 Aug 30]. Available at: http://mathworld.wolfram.com/FunctionCentroid.html.

[41] Li J, John R, Coupland S, et al. On nie-tan operator and type-reduction of interval type-2 fuzzy sets. IEEE Trans Fuzzy Syst [Internet]. 2018 [cited 2019 Aug 30]; 26:1036–1039. Available at: https://ieeexplore.ieee.org/document/7849207/.

[42] Passino KM, Yurkovich S. Fuzzy control. Addison-Wesley; 1998.

[43] Liang Q, Mendel JM. Interval type-2 fuzzy logic systems: theory and design. IEEE Trans Fuzzy Syst [Internet]. 2000 [cited 2019 Aug 30]; 8:535–550. Available at: http://ieeexplore.ieee.org/document/873577/.

[44] Roose AI, Yahya S, Al-Rizzo H. Fuzzy-logic control of an inverted pendulum on a cart. Comput Electr Eng [Internet]. 2017 [cited 2019 Aug 30]; 61:31–47. Available at: https://www.sciencedirect.com/science/article/pii/S0045790617313654.

[45] Ashraf Z, Roy ML, Muhuri PK, et al. Interval type-2 fuzzy logic system based similarity evaluation for image steganography. Heliyon. May 2020;6(5):e03771. https://doi.org/10.1016/j.heliyon.2020.e03771.

[46] Shukla AK, Yadav M, Kumar S, et al. Veracity handling and instance reduction in big data using interval type-2 fuzzy sets. Eng Appl Artifi Intell. 2020;88:103315. https://doi.org/10.1016/j.engappai.2019.103315.

[47] Cuevas F, Castillo O, Cortes-Antonio P. Optimal Design of Interval Type-2 Fuzzy Tracking Controllers of Mobile Robots Using a Metaheuristic Algorithm. In: Melin P, Castillo O, Kacprzyk J, editor. Recent Advances of Hybrid Intelligent Systems Based on Soft Computing. Studies in Computational Intelligence, vol 915. Cham: Springer; 2021. https://doi.org/10.1007/978-3-030-58728-4_18

[48] Chen C, Wu D, Garibaldi JM, et al. A Comprehensive study of the efficiency of type-reduction algorithms. IEEE Trans Fuzzy Syst. March 2020. doi:10.1109/TFUZZ.2020.2981002.

[49] https://www.mathworks.com/matlabcentral/fileexchange/66578-interval-type-2-fuzzy-controller-simple-implementation

[50] Wu D, Mendel JM. Enhanced Karnik–Mendel algorithms. IEEE Trans Fuzzy Syst. 2009;17(4): 923–934.

**Appendix 1**

| Abbreviation | Meaning |
|--------------|---------|
| CoA          | Centre of area |
| FOU          | Footprint of uncertainty |
| GC           | Geometric centroid |
| IT2 FLS      | Interval type-2 fuzzy logic system |
| IT2 FS       | Interval type-2 fuzzy set |
| IT2 TSK FLC  | Interval type-2 Takagi–Sugeno–Kang fuzzy logic controller |
| KM           | Karnik-Mendel algorithms |
| LMF          | Lower membership function |
| MF           | Membership function |
| NT           | Nie-Tan operator |
| T1           | Type-1 |
| T2           | Type-2 |
| T1 FS        | Type-1 fuzzy set |
| T2 FS        | Type-2 fuzzy set |
| TR           | Type-reduction |
| UMF          | Upper membership function |
| Symbol | Meaning |
|--------|---------|
| $\mathcal{A}$ | Type-2 fuzzy set |
| $A_l$ | The area under the LMF of the output IT2 FS $\tilde{B}(y)$ |
| $A_u$ | The area under the UMF of the output IT2 FS $\tilde{B}(y)$ |
| $b_k$ | The locations of the output's membership functions. |
| $C_l$ | The geometric centre of the LMF of the output IT2 FS $\tilde{B}(y)$ |
| $C_u$ | The geometric centre of the UMF of the output IT2 FS $\tilde{B}(y)$ |
| $\delta$ | The Dirac delta function (used to represent the fuzzy singleton) |
| $F^{\mu}_{ij}(x)$ | The firing set (firing level) of the $k^{th}$ (the $ij^{th}$) rule |
| $f^{\mu}_{ij}$ | The lower firing degree of the $k^{th}$ (the $ij^{th}$) rule |
| $\bar{f}^{\mu}_{ij}$ | The upper firing degree of the $k^{th}$ (the $ij^{th}$) rule |
| $R^k$ or $R^{ij}$ | The $k^{th}$ rule of the rule base |
| $\mathcal{T}(\cdot)$ and $\star$ | The T-norm operator (commonly the minimum or product) |
| $\mu_{F^{\mu}_{ij}}(x)$ | The upper membership grade of $F^{\mu}_{ij}(x)$ |
| $\mu_{F^{\mu}_{ij}}(x)$ | The lower membership grade of $F^{\mu}_{ij}(x)$ |
| $\mu_{B}(y)$ | The membership grade of the IT2 FS $\tilde{B}$ |
| $\bar{\mu}_{B}(y)$ | The lower membership function LMF($\tilde{B}$) |
| $\mu_{B}(y)$ | The upper membership function UMF($\tilde{B}$) |
| $\alpha_L, \alpha_U$ | The scaling factor for the approximated LMF and UMF |
| $\sigma_L, \sigma_U$ | The standard deviation of the approximated LMF and UMF |
| $\sqcup$ | The join operation |