Information Sciences 181 (2011) 2933–2950

Contents lists available at ScienceDirect

Information Sciences

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

Interval Type-2 fuzzy voter design for fault tolerant systems

Ondrej Linda *, Milos Manic

Computer Science Dept., University of Idaho, USA

ARTICLE INFO

Article history:
Received 6 August 2010
Received in revised form 14 March 2011
Accepted 20 March 2011
Available online 3 April 2011

Keywords:
Fuzzy voting scheme
Interval Type-2 fuzzy logic
Fault tolerant systems

ABSTRACT

A voting scheme constitutes an essential component of many fault tolerant systems. Two types of voters are commonly used in applications of real-valued systems: the inexact majority and the amalgamating voters. The inexact majority voter effectively isolates erroneous modules and is capable of reporting benign outputs when a significant disagreement is detected. However, an application specific voter threshold must be provided. On the other hand, amalgamating voter, such as the weighted average voter, reduces the influence of faulty modules by averaging the input values together. Unlike the majority voters, amalgamating voters are not capable of producing benign outputs. In the past, a Type-1 (T1) fuzzy voting scheme was introduced, allowing for both smooth amalgamation of voter inputs and effective signalization of benign outputs. The presented paper proposes an extension to the fuzzy voting scheme via incorporating Interval Type-2 (IT2) fuzzy logic. The IT2 fuzzy logic allows for an improved handling of uncertain assumptions about the distributions of noisy and erroneous inputs which are essential for correct design of the fuzzy voting scheme. The proposed voter design features robust performance when the uncertainty assumptions dynamically change over time. The IT2 fuzzy voter architecture was compared against the average voter, inexact majority voter, and the T1 fuzzy voter using a refined experimental harness. The reported results demonstrate improved availability, safety and reliability of the presented IT2 fuzzy voting scheme.

1. Introduction

Voting schemes constitute fundamental components of many fault tolerant systems [30]. Voters provide error masking and noise removal capabilities in a variety of real world applications [18]. Voting can be implemented at different system levels, for example in data fusion at sensory level using a set of redundant sensors [24], in control systems, where redundant HW or SW components are used to produce control signals [39], or at the actuator level [31].

Voting systems typically include an arbiter that fuses the inputs from multiple redundant modules. This manuscript focuses on the Triple-Modular Redundancy (TMR) system. A schematic view of the TMR system is depicted in Fig. 1. Various approaches for arbitration, such as amalgamating or majority voting, can be found in literature [18]. The amalgamating voters use the weighted average to obtain the final output value. The assumptions of independent noise and error signals in the system are used to reduce their impact on the fused outputs. On the other hand, the majority voter attempts to isolate erroneous inputs by excluding them from the arbitration process [17]. In case of real valued systems, inexact majority voting must be used. However, such systems require an application specific distance threshold for evaluating the equality among voters. Unlike the amalgamating voters, the majority voters produce benign outputs in case of a complete disagreement

* Corresponding author. Address: Computer Science Dept., University of Idaho, 1776 Science Center Dr., Idaho Falls, ID 83402, USA. Tel.: +1 208 227 3919.
E-mail addresses: olinda@uidaho.edu (O. Linda), misko@ieee.org (M. Manic).

0020-0255/$ - see front matter
doi:10.1016/j.ins.2011.03.008

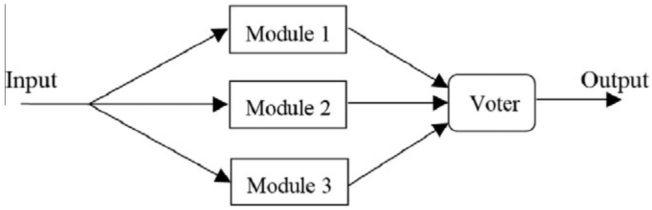

Fig. 1. A Triple-Modular Redundancy (TMR) voting system [19].

among voter inputs. Here, benign output refers to a complete disagreement among individual modules likely leading to an incorrect result of the arbitration.

Various applications of fuzzy logic related to voting and group decision-making systems can be found in literature. A fuzzy extension to the equivalence relation was proposed in [22], where fuzzy logic was used to smooth out the precise equivalence concept. In a similar manner, the smoothing of the output decisions provided by fuzzy logic was used for combining of the results of boosting classifiers and pattern classification algorithms [13,16]. Recently, a consensus model for group decision-making utilizing fuzzy preference relations and fuzzy linguistic context was also developed [12,25]. A distinct approach to group decision-making utilizing the ranking values computed based on IT2 fuzzy sets was presented by Chen et al. [3].

A Type-1 (T1) fuzzy voting scheme was proposed in the past [19]. This system utilized T1 fuzzy logic to combine the amalgamating properties of averaging voters with the error separation capabilities of majority voters. Fuzzification of voter inputs eliminated the need for specifying a precise and application-dependent distance threshold, while maintaining the ability of the voter to report benign outputs. In [19], the experimental results demonstrated that the T1 fuzzy voter offers increased availability and reduces the number of benign outputs at a cost of minor degradation in the safety performance.

This manuscript extends the T1 fuzzy voting design via incorporating the Interval Type-2 (IT2) fuzzy logic [14,27]. First, it is demonstrated that the behavior of the T1 fuzzy voting scheme relies on an accurate assessment of the expected distributions of noise and errors in the system. The performance of the T1 fuzzy controller is susceptible to dynamic uncertainties manifested as variations of the assumed input signals distributions. The IT2 fuzzy logic has the potential to cope with such dynamic uncertainties. In other words, IT2 fuzzy logic based voters improve the system quality and stability when the initial assumptions about the system behavior change over time.

Next, a specific asymmetrical blurring scheme for obtaining an IT2 fuzzy voting design from a given T1 fuzzy voter architecture is proposed. The performance of the IT2 fuzzy voting scheme is demonstrated on a refined experimental harness. The experimental system allows for a separate modeling of noise and error signal distributions. The IT2 fuzzy voting system was compared against the T1 fuzzy voter, the inexact majority voter, and the simple averaging voter.

The rest of the paper is organized as follows. Section 2 provides a background overview on T1 and IT2 fuzzy logic. Section 3 reviews and analyses the previously published T1 fuzzy voting system. The novel presented IT2 fuzzy voting scheme is described in Section 4. Experimental results are presented in Section 5 and the paper is concluded in Section 6.

2. Overview of T1 and IT2 fuzzy logic systems

This section provides an overview of T1 and IT2 fuzzy logic systems.

2.1. Type-1 fuzzy logic systems

T1 fuzzy logic systems (FLSs) have been used in a wide range of engineering applications [26]. The main advantage of T1 FLSs is the ability to incorporate human-understandable knowledge encoded as linguistic fuzzy rules. In addition, T1 FLSs can cope with ambiguity, imprecision and uncertainty.

A T1 FLS is composed of four major components – input fuzzification, fuzzy inference engine, fuzzy rule base and output defuzzification. The considered Mamdani FLS maintains a fuzzy rule base populated with fuzzy linguistic rules in an implicative form. As an example, consider rule R_k:

$$\text{Rule } R_k : \textbf{IF } x_1 \text{ is } A_1^k \textbf{ AND} \cdots \textbf{AND } x_n \text{ is } A_n^k \textbf{ THEN } y_k \text{ is } B^k \tag{1}$$

Here, symbol A_j^k and B^k denote the jth input fuzzy set and the output fuzzy set, n is the dimensionality of the input vector $\vec{x}$, and y_k is the associated output variable. Each element of the input vector $\vec{x}$ is first fuzzified using the respective fuzzy membership function (e.g. Gaussian, triangular, trapezoidal, etc.). The fuzzification of input x_i into fuzzy set A_i^k yields a fuzzy membership grade $\mu_{A_i^k}(x_i)$. Using the minimum t-norm, the degree of firing of rule R_k can be calculated as

$$\mu_{R_k}(\vec{x}) = \min\{\mu_{A_i^k}(x_i)\}, \quad i = 1, \ldots, n \tag{2}$$

Upon applying the rule firing strength via the t-norm operator (e.g. minimum or product) to each rule consequent, the output fuzzy sets are aggregated using the t-conorm operator (e.g. the maximum operator), resulting in an output fuzzy set B. Detailed description of the fuzzy inference process can be found in [26].

Finally, the output fuzzy set B is defuzzified in order to obtain the final output value. Several defuzzification techniques can be found in literature, e.g. centroid defuzzifier, center-of-sums defuzzifier, heights defuzzifier, or the center-of-sets defuzzifier [26]. Here, upon discretizing the output domain into N samples, the centroid defuzzifier is used to produce a final output value y:

$$y = \frac{\sum_{i=1}^{N} y_i \mu_B(y_i)}{\sum_{i=1}^{N} \mu_B(y_i)} \tag{3}$$

2.2. Interval Type-2 fuzzy logic systems

Dynamic uncertainties, ubiquitous in many real world applications, can negatively affect the performance of the implemented control system [1,4,26]. The T1 FLS uses fixed fuzzy memberships that cannot directly address such variable conditions. Neglecting this uncertainty can lead to a subsequent deterioration of the system quality [10].

Type-2 fuzzy logic systems (T2 FLSs), originally proposed by Zadeh [40], have recently gained increased attention [6,14,27,28,35,41]. Their ability to outperform T1 FLSs when presented with dynamic uncertainties was demonstrated in many engineering applications [10,20,26]. T2 FLSs allow for modeling of dynamic uncertainties by introducing additional degrees of freedom (the T2 fuzzy membership functions use membership degrees that are themselves fuzzy sets).

Recently, new representations for general T2 FLSs such as geometric T2 FLSs [6], *zSlices* [35], α-planes [21,28] or α-cuts [11] were introduced. However, the Interval T2 (IT2) FLSs [27,37], also known as Interval-Valued FLSs [7,32,33] are still the most commonly used. The favored computational efficiency of IT2 FLSs comes from using the concept of Footprint of Uncertainty (FOU) for defining each IT2 fuzzy set. The FOU can be described by two bounding T1 fuzzy membership functions [26].

An IT2 fuzzy set $\widetilde{A}$ can be expressed as [26]:

$$\widetilde{A} = \int_{x\in X} \int_{u\in J_x} 1/(x,u) \quad J_x \subseteq [0,1] \tag{4}$$

Here, x and u are the primary and the secondary variables, and J_x is the primary membership of x. In case of IT2 fuzzy sets, all secondary grades of fuzzy set $\widetilde{A}$ are equal to 1. A vertical slice of the IT2 fuzzy set can be obtained by instantiating the variable x into a specific value x':

$$\mu_{\widetilde{A}}(x = x', u) \equiv \mu_{\widetilde{A}}(x') = \int_{u\in J_{x'}} 1/u \quad J_{x'} \subseteq [0,1] \tag{5}$$

The domain of the primary memberships J_x defines the FOU of fuzzy set $\widetilde{A}$:

$$FOU(\widetilde{A}) = \bigcup_{x\in X} J_x \tag{6}$$

The FOU of an IT2 fuzzy set $\widetilde{A}$ can be also described by its upper and lower membership functions. Hence

$$FOU(\widetilde{A}) = \bigcup_{\forall x\in X} (\underline{\mu}_{\widetilde{A}}(x), \bar{\mu}_{\widetilde{A}}(x)) \tag{7}$$

For an ease of understanding, the concepts of FOU and the lower and upper membership function are depicted in Fig. 2. This representation offers a substantial simplification when compared to general T2 FLSs. Here, only two T1 fuzzy membership functions (the upper, $\bar{\mu}_{\widetilde{A}}(x)$, and the lower, $\underline{\mu}_{\widetilde{A}}(x)$ boundary of the FOU) are used to completely describe each IT2 fuzzy set. This simplification is then transferred through an altered inference mechanism utilizing the modified interval T2 fuzzy join and meet operations [26]. The interval join and meet operations work exclusively with the FOU of the IT2 fuzzy sets, thus removing much of the computational burden that comes with general T2 fuzzy set arithmetic.

In order to obtain an output value, the resulting output IT2 fuzzy set $\widetilde{B}$ must be first type-reduced and then defuzzified. The interval centroid of the IT2 fuzzy set $\widetilde{B}$ is defined as [15]:

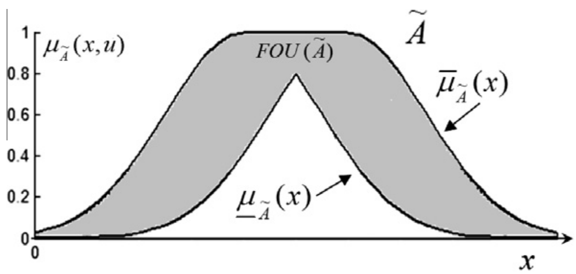

Fig. 2. Interval Type-2 fuzzy set.

$$C_{\tilde{B}} = \int_{\theta_1 \in J_{x_1}} \cdots \int_{\theta_N \in J_{x_N}} 1 \Bigg/ \frac{\sum_{i=1}^{N} x_i \theta_i}{\sum_{i=1}^{N} \theta_i} \quad (8)$$

Here, every combination of variables $\theta_1, \ldots, \theta_N$, each selected from the respective primary memberships, forms and embedded fuzzy set [26]. In the presented manuscript, the Nie–Tan (N–T) type reduction method was used due to its computational inexpensiveness [29]. The N–T method first computes the centroid of each vertical slice in each primary membership J_{x_i} of the output IT2 fuzzy set and then defuzzifies the resulting type-reduced set. For the IT2 fuzzy sets, the centroid u_j of the jth vertical slice simply becomes

$$u_j = \frac{1}{2}(\underline{\mu}_{\tilde{A}}(x_j) + \bar{\mu}_{\tilde{A}}(x_j)) \quad (9)$$

Assuming that the output domain was discretized into M samples, the defuzzified output is obtained as in [29]:

$$y = \frac{\sum_{j=1}^{M} x_j \bar{\mu}_{\tilde{A}}(x_j) + \sum_{j=1}^{M} x_j \underline{\mu}_{\tilde{A}}(x_j)}{\sum_{j=1}^{M} \bar{\mu}_{\tilde{A}}(x_j) + \sum_{j=1}^{M} \underline{\mu}_{\tilde{A}}(x_j)} \quad (10)$$

In other literature, the Karnik–Mendel (KM) iterative procedure is typically used [38]. The KM method computes the left and the right boundaries of the output interval centroid $C_{\tilde{B}}$. However, since this information is not utilized in the presented work, the computationally less expensive N–T method was favored instead. Other available methods for defuzzification of IT2 FS are the direct defuzzifier [5], the collapsing method [9] or the sampling approach [8].

3. Type-1 fuzzy voting scheme: review and analysis

This section reviews the previously published T1 fuzzy voting scheme [19]. In addition, an analysis of the T1 fuzzy voter is presented.

3.1. Type-1 fuzzy voting scheme

A T1 fuzzy voting scheme was introduced by Latif-Shabgahi and Hirst in [19]. This voter design combined the amalgamating capability of averaging voters with the benign input and outlier recognition ability of inexact majority voters. The T1 fuzzy voting architecture was explained on an example of a TMR system (illustrated in Fig. 1). Nevertheless, the proposed concept can be generalized for an n-modular system. An overview of this T1 fuzzy voter architecture is shown in Fig. 3. The fuzzy voter computes the weighted average of the system inputs using weights, which are determined by a T1 fuzzy inference engine.

In the first step of the algorithm, the differences between all pairs of input signals are computed as follows:

$$d_{ij} = |x_i - x_j|, \quad i \neq j \quad (11)$$

Here, x_i and x_j are distinct voter inputs.

Next, the differences d_{ij} are fuzzified using the fuzzy difference concept $\mu_A(d_{ij})$, where $A : \{Small, Medium, Large\}$. As shown in Fig. 4, the design of the triangular fuzzy difference membership functions is determined by three parameters p, q and r. As denoted in [19], the behavior of the fuzzy voter can be tuned by adjusting these parameters. Fig. 4 also depicts that the fuzzy difference concept divides the range of difference values into three sub-regions of *definite agreement*, *uncertain fuzzy agreement* and *definite disagreement*. The second part of this section demonstrates that the correct initial assumptions about the distributions of noise and errors in the voter's inputs are required for the selection of parameters p, q and r.

The computed fuzzy differences describe the agreement of each input signal of the TMR system with the other two inputs. The fuzzy voter transforms this input measure into a concept of fuzzy agreeability. In [19], the fuzzy agreeability concept $\mu_B(w_i)$ of weight w_i was modeled using five evenly positioned output fuzzy sets $\{vlow, low, med, high, vhigh\}$. As depicted in Fig. 5, the output fuzzy sets are determined by three parameters u, v and w, which allow for additional tuning of the behavior of the voter.

The mapping of fuzzy differences into fuzzy agreeability is achieved via standard fuzzy inference mechanism [26]. The fuzzy knowledge base contains linguistic rules in the implicative form given in (1). The minimum t-norm was used to

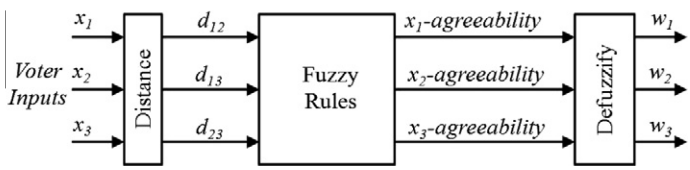

Fig. 3. Architecture of the T1 fuzzy voter [19].

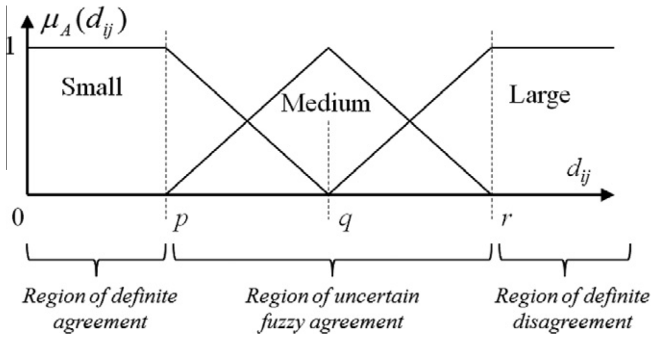

Fig. 4. Fuzzy difference concept of T1 fuzzy voter.

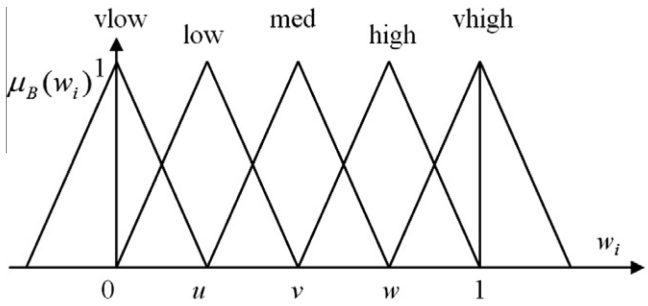

Fig. 5. Fuzzy agreeability concept of T1 fuzzy voter [19].

Table 1
Rule matrix for fuzzy input variables [19].

&		d_{ij}		
		Small	Medium	Large
d_{ik}	Small	Vhigh	High	Med
	Medium	High	Low	Vlow
	Large	Med	Vlow	Vlow

compute the rule firing strength based on the set of antecedents. The outputs of the fuzzy rules were aggregated using the maximum t-conorm.

The set of linguistic fuzzy rules provides a fundamental description of the voting behavior. Table 1 shows the fuzzy rule base for a TMR system adopted from [19] and also used in this paper. The included rules provide an intuitive way of mapping the fuzzy differences between individual inputs to the fuzzy agreeability concept. In this specific set of linguistic rules, higher value of fuzzy agreeability is assigned when smaller differences between inputs are detected. In a similar manner, the system outputs small fuzzy agreeability when the input value is far apart from the other inputs. Other variations of the fuzzy rule table are also possible.

The fuzzy voting scheme uses the weighted average calculation to obtain the final output. The weights are determined by defuzzifying the fuzzy agreeability concept of each input signal. In [19], the height-type defuzzifier was used. For a set of output fuzzy sets B_j, as depicted in Fig. 5, and the calculated centroids C_j of output fuzzy sets B_j the weight w_i of input x_i can be computed as

$$w_i = \frac{\sum_{j=1}^{M} \mu_{B_j} C_j}{\sum_{j=1}^{M} \mu_{B_j}} \tag{12}$$

Here, M is the number of output fuzzy sets. For the special case depicted in Fig. 5, $C_{vlow} = 0$, $C_{low} = u = 0.25$, $C_{medium} = v = 0.5$, $C_{high} = w = 0.75$ and $C_{vhigh} = 1$.

The final output value y for an m-way fuzzy voter is obtained by weighting each input signal x_i with the calculated weight w_i:

$$y = \frac{\sum_{i=1}^{m} x_i w_i}{\sum_{i=1}^{m} w_i} \tag{13}$$

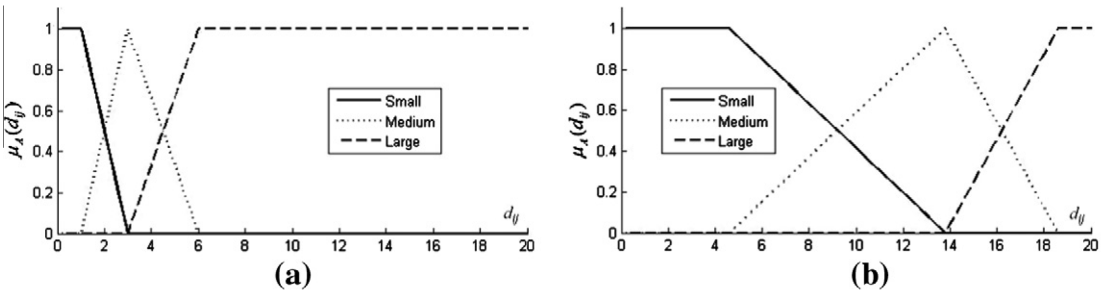

Fig. 6. Fuzzy difference concept D_1 (a) and D_2 (b).

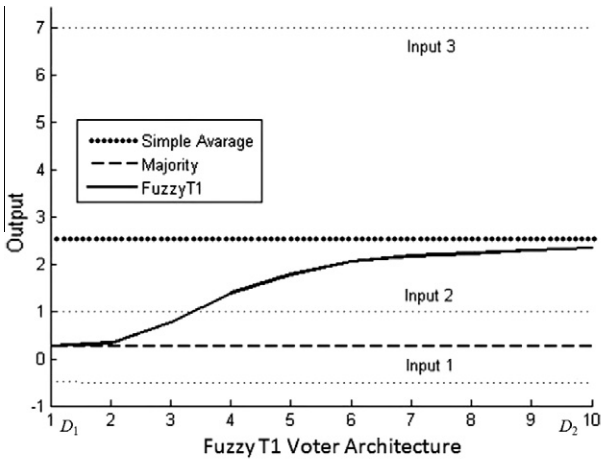

Fig. 7. Output of the T1 fuzzy voter for different system parameters.

The T1 fuzzy voting system is specified by its six parameters $\{p,q,r,u,v,w\}$. Two common approaches can be adopted to set the values of these parameters. Typically, the parameters can be set based on expert's knowledge about the problem domain. Alternatively, the parameters can be tuned using an optimization algorithm to maximize specific performance criteria. Here, for example the genetic algorithms can be utilized [23].

3.2. T1 fuzzy voter analysis

In the original publication [19], no guidelines were given for determining the parameters of the T1 fuzzy voting system. In this section, it is demonstrated that appropriate choice of these parameters is necessary, should the fuzzy voter perform as desired.

Consider a set of 10 distinct fuzzy voters obtained by linearly interpolating between two fuzzy difference concepts D_1 and D_2, with parameter values as follows (depicted in Fig. 6):

$$D_1 : p = 1,\ q = 3,\ r = 6 \tag{14}$$

$$D_2 : p = 5,\ q = 14,\ r = 19 \tag{15}$$

This interpolation process constructs 10 fuzzy voters by linearly shifting (stretching) the fuzzy difference sets of D_1 towards the fuzzy difference sets of D_2. For instance, voter number 5, will be constructed using the parameters $p = 3$, $q = 8.5$, $r = 12.5$.

Consider three inputs to the TMR fuzzy voter system with values $Input_1 = -0.5$, $Input_2 = 1.0$ and $Input_3 = 7.0$, when the correct input value is assumed to be 0. Signals $Input_1$ and $Input_2$ simulate noisy inputs, while signal $Input_3$ carries an erroneous value.

Fig. 7 demonstrates the output of the 10 tested T1 fuzzy voters obtained by interpolating between the fuzzy difference concepts D_1 and D_2. It can be observed that for fuzzy difference concept D_1 the T1 fuzzy voter behaves similarly as the inexact majority voter. As the interpolated fuzzy difference concept approaches concept D_2, the behavior of the fuzzy voter smoothly shifts towards the performance of the averaging voter. This change in behavior is achieved by assigning more fuzzy agreeability to higher values of fuzzy difference between the input signals. This result demonstrates that for a given distribution of input signals, the behavior of the T1 fuzzy voter can be tuned by changing the design of the fuzzy difference concept.

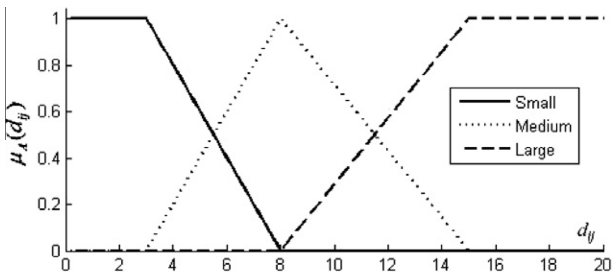

Fig. 8. Fuzzy difference concept D_3.

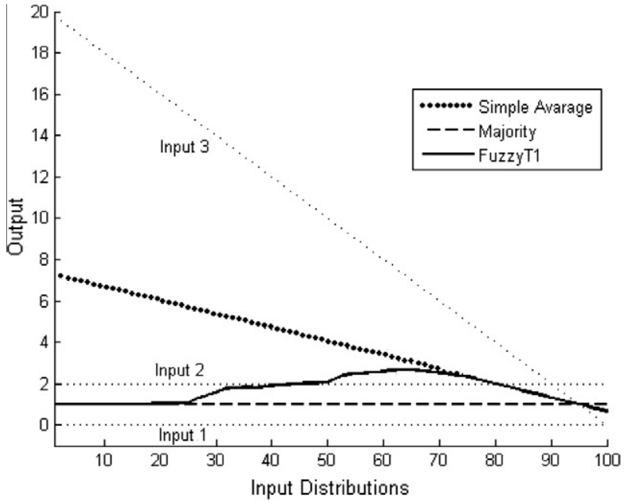

Fig. 9. Output of the fuzzy voter for different input values.

The opposite problem can be encountered when the parameters of the T1 fuzzy voter are fixed, but the distributions of noise and errors in the input signals dynamically change over time (simulating dynamic uncertainties). Consider the fuzzy difference concept D_3, depicted in Fig. 8 and defined as

$$D_3 : p = 3,\ q = 8,\ r = 15 \tag{16}$$

The behavior of this TMR T1 fuzzy voter is demonstrated for three input signals, where $Input_3$ signal is increasing its difference from the other two inputs. Signals $Input_1$ and $Input_2$ are constant. This scenario demonstrates dynamically changing error signal distribution imposed on the $Input_3$ signal. The behavior of the T1 fuzzy voter using the fuzzy difference concept D_3 is demonstrated in Fig. 9. It can be observed that for the chosen parameters, the final output of the T1 fuzzy voter behaves similar to the averaging voter for small differences between inputs and it smoothly approaches the behavior of the majority voter for larger input signal differences. This behavior variation is achieved by shifting the difference of the $Input_3$ signal away from the averaging section (region of *definite agreement* in Fig. 4) of the fuzzy voter towards the outlier detection section (region of *definite disagreement* in Fig. 4).

The presented analysis revealed the importance of designing the T1 fuzzy voter architecture respective of the expected distributions of noise and error signals in voter inputs. Furthermore, it was observed that the T1 fuzzy voter behavior is highly sensitive to dynamically changing and uncertain input noise and error distributions. In order to reduce such sensitivity and improve the robustness of the voting system, the IT2 fuzzy voting scheme is introduced in this paper.

4. Interval Type-2 fuzzy voting scheme

The performance of the T1 fuzzy voting scheme deteriorates when confronted with dynamically changing and uncertain distributions of noise and errors. Many researchers showed that T2 fuzzy logic controllers are more robust to uncertain inputs and that they can cope with dynamic uncertainty [10,20,26]. In this section, an IT2 fuzzy voting scheme is proposed.

Assume a previously constructed T1 fuzzy voter architecture specified by its six design parameters $\{p, q, r, u, v, w\}$ and by a set of linguistic fuzzy rules as depicted in Figs. 4, 5 and shown in Table 1. The T1 fuzzy membership functions can be transformed into their respective IT2 counterparts by introducing the secondary uncertainty domain and blurring the fuzzy sets into their FOUs. In this manner, the T1 fuzzy concepts of difference and agreeability become IT2 fuzzy concepts. The created IT2 fuzzy voting scheme then uses the IT2 fuzzy inference process.

In the specific application of fuzzy voting, the introduction of IT2 fuzzy membership functions improves the ability of the fuzzy voter to cope with varying noise distributions in the input signals. However, a special care needs to be applied when blurring the fuzzy difference membership functions. As already argued in [19], the range of input differences is divided by the fuzzy sets into three distinct regions of *definite agreement*, *uncertain fuzzy agreement* and *definite disagreement* (Fig. 4). Small values of input difference are mapped into the region of *definite agreement*, where the voter performs mainly the averaging function. Next, the middle region of *uncertain fuzzy agreement* considers the inputs to be correct to a lesser or greater extent. Finally, the third region of *definite disagreement* is responsible for recognizing and isolating erroneous inputs. The averaging capability of the fuzzy voter should be strengthened in the first two regions. However, such modification should not affect the region of *definite disagreement*, where strict recognition of the erroneous inputs and their subsequent isolation is desired.

Here, an asymmetric blurring scheme of the input fuzzy sets of the fuzzy difference concept is presented. In the proposed design, the *Small* difference fuzzy set is blurred symmetrically in both directions around the T1 fuzzy membership function. The T1 fuzzy membership function becomes the principal membership function of the created IT2 fuzzy set. The *Medium* difference fuzzy set is blurred only in the direction of the smaller differences. Finally, the *Large* difference fuzzy set remains unchanged as T1 fuzzy set. In this manner, the IT2 fuzzy logic improves the robustness of the controller in the first two regions of *definite agreement* and *uncertain fuzzy disagreement*. However, the ability of the controller to recognize faulty inputs in the *definite disagreement* region is maintained by using the T1 fuzzy sets for the fuzzy difference concept *Large*. The output fuzzy agreeability fuzzy sets are blurred symmetrically around the respective output T1 fuzzy sets. In this manner, uncertainty about the output value is incorporated into the controller's design.

The IT2 fuzzy voter design process goes as follows. First, the proposed design methodology computes the maximum amount of allowable blur for the provided T1 fuzzy voting architecture. This process is depicted in Fig. 10. The calculation of maximum allowable blur ensures that reasonable amount of overlap between neighboring fuzzy sets is achieved. The maximum allowable blur for fuzzy difference sets *Small* and *Medium* Δ_X^S and Δ_X^M and the output fuzzy sets of fuzzy agreeability Δ_{Out} are determined as follows:

$$\left\{\Delta_X^S, \Delta_X^M, \Delta_{Out}\right\} = \left\{\frac{(q-p)}{3}, \frac{2(q-p)}{3}, \frac{(v-u)}{4}\right\} \tag{17}$$

The computation of the maximum allowable blur for the output fuzzy sets Δ_{Out} assumes regular distribution of fuzzy agreeability membership functions. If this is not the case, the smallest distance between two neighboring output fuzzy sets can be used instead.

Second, the user-specified blurring constant $\alpha \in [0,1]$is applied to calculate the parameters of the new IT2 fuzzy membership functions. This process is illustrated in Fig. 11(b) and (d) for the fuzzy difference concept. The final design of the membership functions of the IT2 fuzzy voting scheme is demonstrated in Fig. 12.

The size of the FOUs of the IT2 fuzzy sets can be controlled by varying the blurring constant α. Values close to zero result only in minor modification of the original T1 fuzzy voter. Values close to 1 create IT2 fuzzy voting system with large FOUs producing maximum smoothing of the arbitrated result. It should be noted here that other blurring schemes that can lead to different voting behaviors and performance are also possible. Alternatively, optimization method such as genetic algorithms could be applied to tune the IT2 fuzzy voter to the specific operating conditions and the desired objective function [34,36].

The proposed asymmetrical blurring design is in agreement with an experimental study performed by the authors [20]. Here, it was demonstrated that the symmetrical blurring design introduces global smoothing of the control surface. This can improve the ability of the controller to deal with noisy inputs; however, it can also reduce the responsiveness of the controller. In scenarios when fast and sudden response is required (e.g. turning around corners for a wall-following robot, or determining erroneous modules in fault-tolerant systems), the T1 fuzzy membership function could outperform their IT2 counterparts.

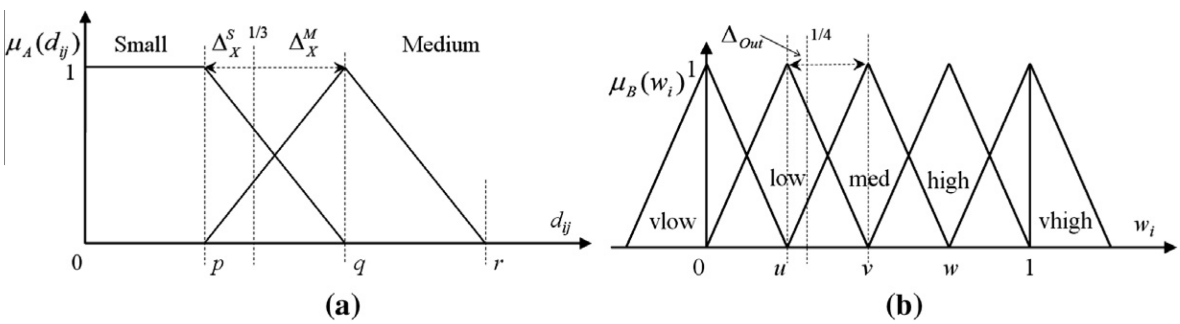

Fig. 10. The proposed blurring of the *Small* and *Medium* fuzzy difference membership functions (a) and blurring of the output concept of fuzzy agreeability.

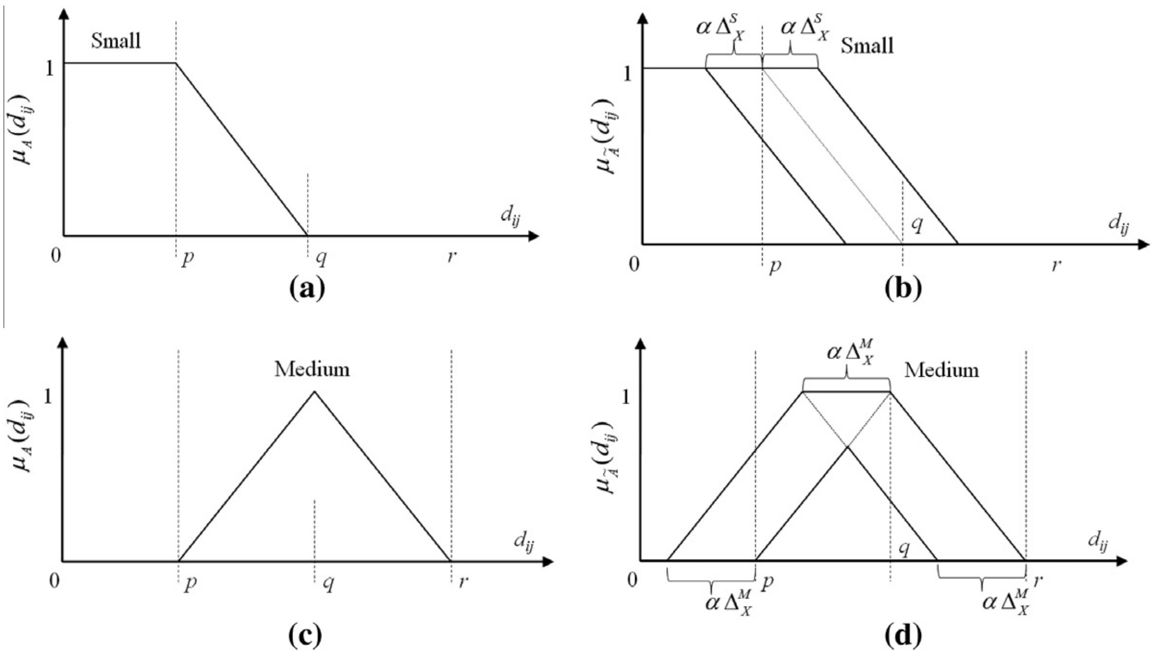

Fig. 11. The proposed blurring of the *small* (a) and (b) and the *medium* (c) and (d) fuzzy difference membership functions.

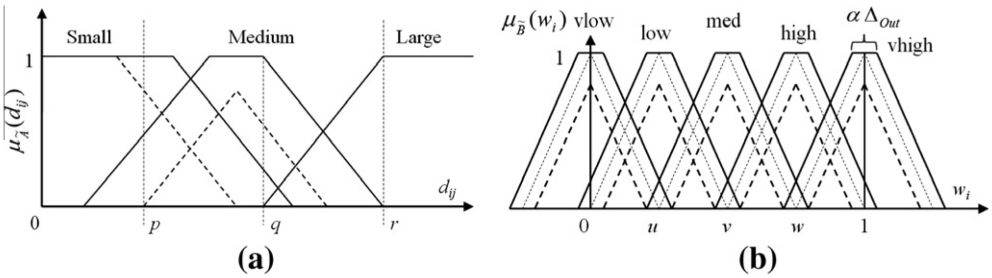

Fig. 12. Type-2 fuzzy difference concept (a) and the type-2 fuzzy agreeability concept (b) obtained by blurring the original Type-1 membership functions.

The IT2 fuzzy voting scheme was explained on an example of a Triple-Modular Redundancy system. However, the algorithm can be easily extended to voting systems with higher dimensionality. For an m-way voter m weights must be calculated, one for each individual input signal. The number of individual fuzzy logic engines will also be m. Each fuzzy logic engine takes as input $(m-1)$ differences to the other input signals. For the considered Mamdani type fuzzy logic controller, the fuzzy rule base of each fuzzy logic engine must contain $n^{(m-1)}$ fuzzy rules, where n is the number of input fuzzy sets. This exponential growth of the size of the fuzzy rule base might be considered a potential bottleneck for the fuzzy voting system with higher dimensionality. However, it should be noted that majority of voting systems are constructed as TMR or as a hierarchy of TMR subsystems.

5. Experimental results

This section presents the performance evaluation of the proposed IT2 fuzzy voting system. First, a novel experimental harness is introduced. The proposed experimental setup allows for separate injection of noise and errors into the input signals. Next, the quality of the proposed voting scheme is validated using an exemplary test case. The performance of the voting system is compared to the averaging, inexact majority and the T1 fuzzy voting systems. Finally, the effect of parameter adjustment and the scalability of the fuzzy voting scheme are studied.

5.1. Testing methodology

An experimental harness for testing the performance of voting systems was presented in [2]. In this paper, the testing methodology is further refined in order to better assess the performance of the proposed voting system with respect to noise and errors contained in the input signal. The proposed experimental harness for a TMR system is depicted in Fig. 13.

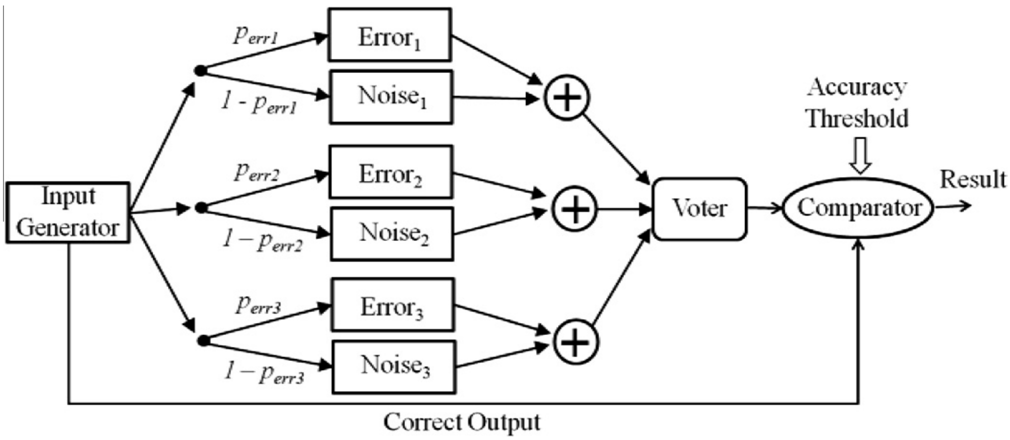

Fig. 13. Refined TMR experimental harness for testing of voting algorithms.

In this scheme, the input generator replicates the input signal into three branches. In each branch, the signal can either undergo an error injection with probability p_{err}, or a noise injection with probability $(1 - p_{err})$. The resulting signals are then presented to the voter, which arbitrates among them. The final result is obtained by comparing the initial correct input value to the voter's output.

When compared to the previous experimental setup presented in [2], the novel methodology allows for separate simulation of erroneous and noisy signals. This is essential in the presented work, since modulating the amplitudes and distributions of noise and error signals simulates the dynamic uncertainties of real world systems. In addition, the likelihood of an error can be adjusted for each module separately.

In the rest of this section an exemplary test case is presented and studied. The outputs of the implemented voters are compared to the correct input signal by using a specified distance threshold. In the performed experiments, the distance threshold should distinguish between noisy and erroneous inputs. Hence, the value of the distance threshold was chosen equivalent to the q parameter of the fuzzy difference concept (Fig. 4). The same value was also selected as the equivalence threshold for the inexact majority voter.

The voter's output is considered correct if its value differs from the correct reference input by lesser value than the distance threshold. In the opposite case, an incorrect voting result is reported. The inexact majority voter and the T1 and the IT2 fuzzy voters can also produce benign outputs. A benign output occurs when a complete disagreement is found among all input signals.

The behavior of a certain voting algorithm can be described by different quality measures. The following values are computed during the experiment: n_c is the number of correct outputs, n_{ic} is the number of incorrect voting outputs, n_b is the number of benign outputs and N is the total number of voting cycles. For the purpose of qualitative comparison, several criteria that are essential to fault tolerant systems are considered. The *availability* measure is computed as $1 - n_b/N$, defining the ratio of time when an output is available to the user. The *safety* measure is calculated as $1 - n_{ic}/N$. The *reliability* measure is obtained as n_c/N. In addition, the Root Mean Square Error (RMSE) of all, correct and incorrect voting cycles was computed. The RMSE measures provide additional insight into the degree of correctness or incorrectness of the arbitrated output.

It is worth noting that the *availability* of the averaging voter is not considered, since the voting scheme is not capable of reporting a benign output. The averaging voter always produces an output value regardless of its meaningfulness.

5.2. Test case

In this experimental evaluation, a simulated TMR system receives a reference input signal, which is set to 0. The TMR system has one faulty module, which produces an error with probability p_{err} = 0.2. The error is manifested as a value of 20.0 injected to the reference signal. In addition all modules inject noise into the input signal. The injected noise has a uniform distribution in the interval $[-Amp, Amp]$, where $Amp \in \{1, \ldots, 12\}$. This interval of values covers a reasonable range of noise signals with respect to the error signal.

In this case study, the parameters of the fuzzy difference concept of the T1 fuzzy voter are as follows:

$$D_A : p = 3,\ q = 7, r = 15 \quad (18)$$

The IT2 fuzzy voter is constructed according to the algorithm presented in Section 4. Here, the blurring constant α = 1 was applied implementing the maximum allowable amount of blur into the design of the respective IT2 fuzzy membership functions. Both T1 and IT2 fuzzy difference and fuzzy agreeability concepts are shown in Fig. 14.

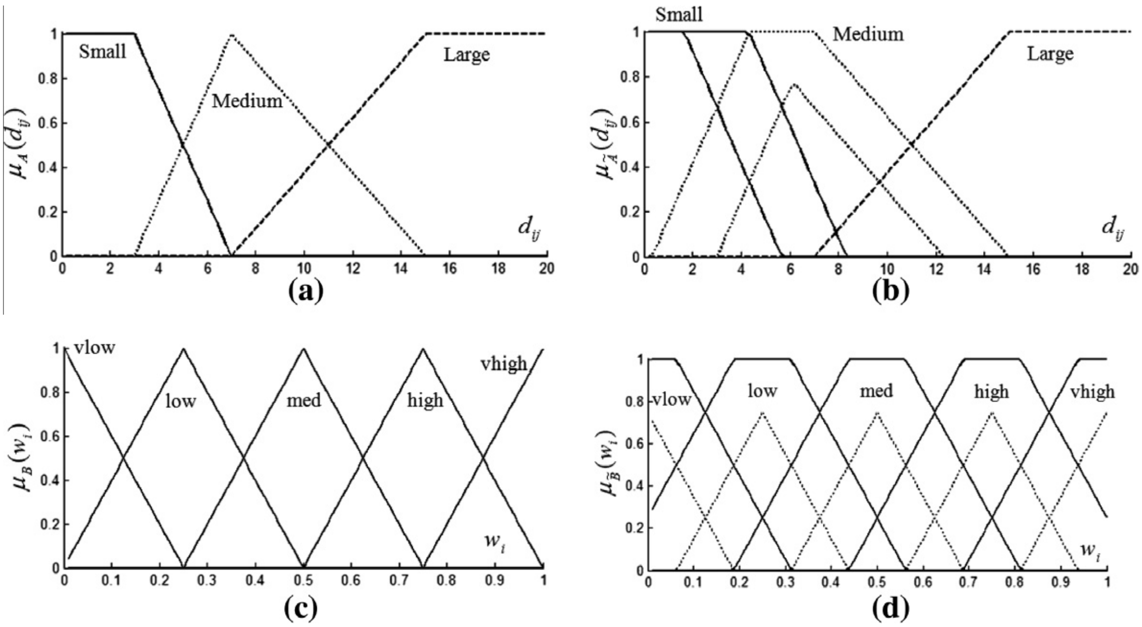

Fig. 14. Fuzzy difference and fuzzy agreeability concepts of the T1 (a) and (c) and IT2 (b) and (d) fuzzy voter for the presented test case.

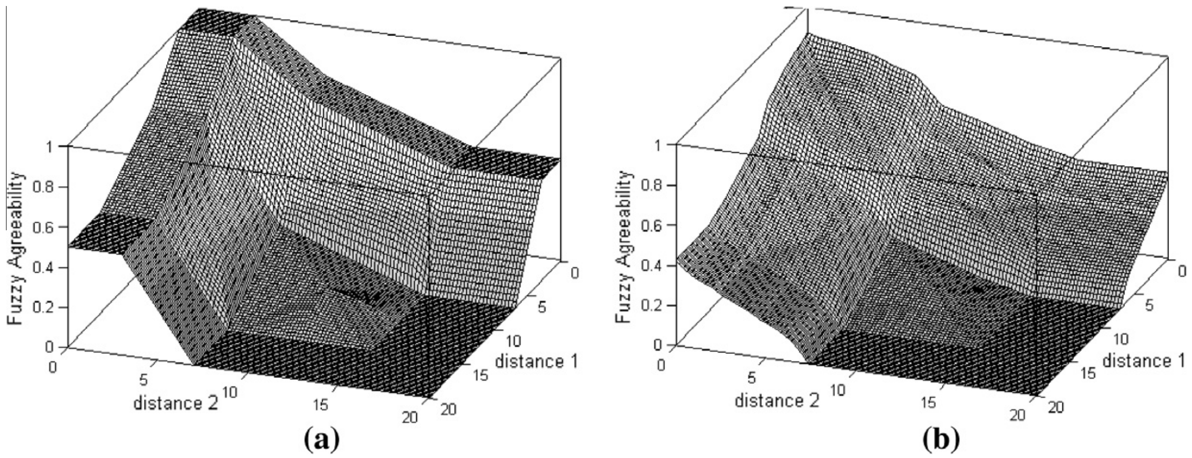

Fig. 15. Comparison of the control surfaces of the T1 (a) and the IT2 (b) fuzzy voters for the presented test case.

The behavior of both TMR T1 and IT2 fuzzy voters can be compared by plotting their respective control surfaces. The control surfaces are constructed by calculating the fuzzy agreeability value for each combination of the input fuzzy differences. The T1 and IT2 fuzzy control surfaces are depicted in Fig. 15.

The following observations can be made. Firstly, as the plotted control surfaces demonstrate, the IT2 fuzzy voter design introduces substantially smother control surface. However, the complexity of the system with respect to the size of the fuzzy rule base remained the same (no additional fuzzy rules or fuzzy sets were used). Hence, smoother arbitrating can be achieved.

Secondly, the plotted control surfaces in Fig. 15 also show the benefit of the proposed asymmetric blurring design for obtaining the IT2 fuzzy voter architecture. While the control performance was smoothed out in the *definite agreement* and the *uncertain fuzzy agreement* regions of the fuzzy difference concept, the desired separation of erroneous inputs in the *definite disagreement* region was retained.

Next, the proposed IT2 fuzzy voter was compared to the averaging, inexact majority and the T1 fuzzy voting scheme. A batch of 10,000 voting cycles was repeated 10 times for each considered noise level. The distance threshold for both the inexact majority voting and for identifying correct voting outputs was set to 7 (the value of parameter q for the fuzzy difference

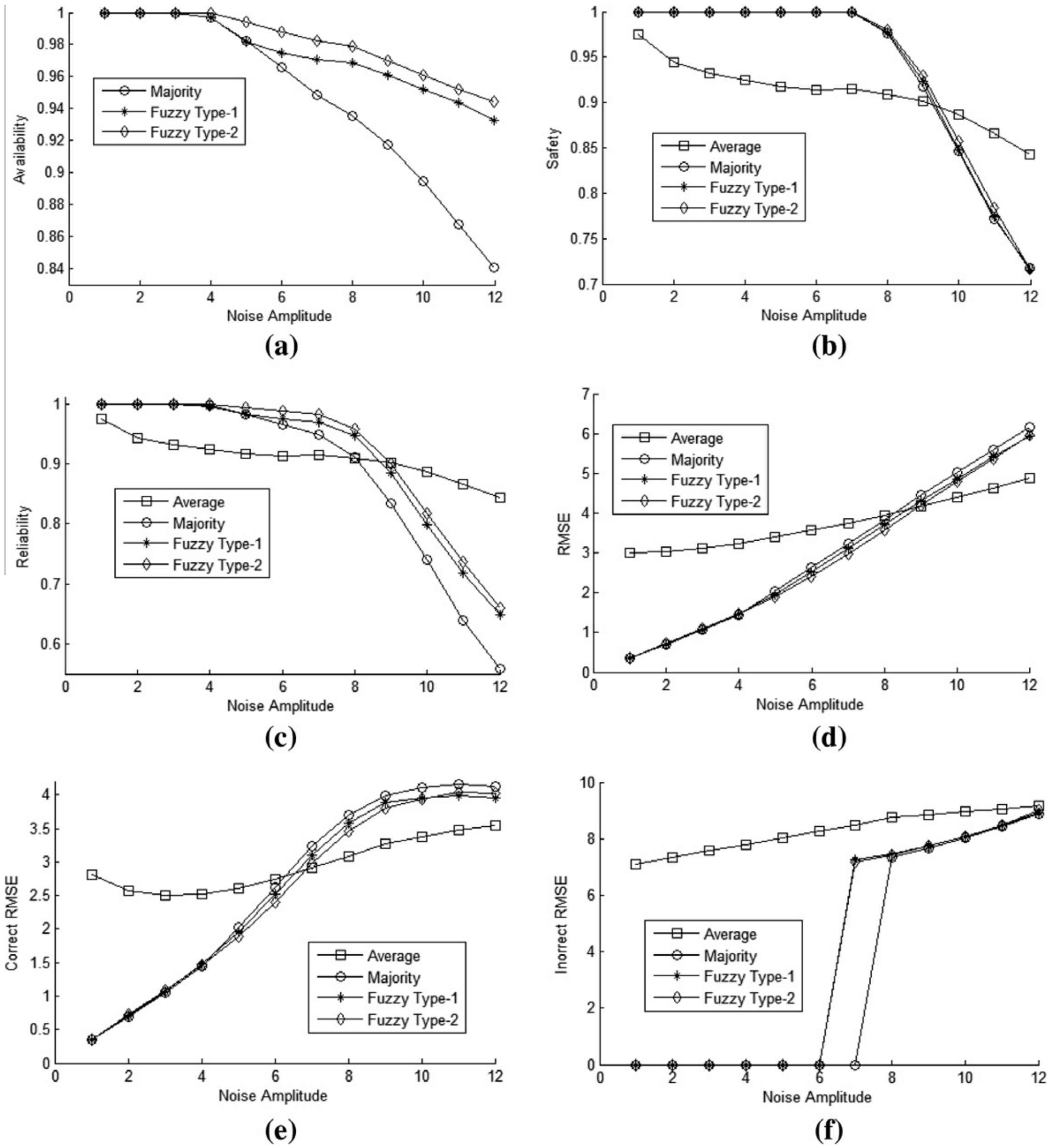

Fig. 16. Experimental results of the presented test case: availability (a), safety (b), reliability (c), RMSE (d), RMSE correct (e), and RMSE incorrect (f).

concept). The average values as well as the standard deviations of the results were computed. The calculated quality measures of *availability*, *safety*, *reliability*, RMSE and RMSE of correct and incorrect voting cycles are all plotted in Fig. 16. In addition, the calculated results are also presented in Tables 2 and 3. Here, both average (bold italics font size) and standard deviations (italics font size) of the results are denoted. For ease of understanding, the best voting performance for each level of noise is highlighted by using bold font in Tables 2 and 3.

By observing Fig. 16 and studying Table 2 it can be concluded that one of the main benefits of the proposed IT2 fuzzy voter is its increased availability. This is achieved by increasing the overlap between neighboring fuzzy sets and thus avoiding premature definite disagreement among system inputs.

Less expected might the fact that the cost of the increased availability was not paid by either decreased safety or reliability. In the matter of fact, in most cases the safety and the reliability measures of the proposed IT2 fuzzy voting scheme were higher than of its T1 counterpart or the inexact majority voter. This can be attributed to the increased overlap between neighboring fuzzy sets and the non-linearity and smoothness of the arbitrating behavior.

Table 2
Comparison of availability, safety and reliability for different voting schemes, mean (bold italic font) and standard deviation (italic font).

Noise amplitude	Availability			Safety				Reliability			
	Majority	Fuzzy T1	Fuzzy IT2	Average	Majority	Fuzzy T1	Fuzzy IT2	Average	Majority	Fuzzy T1	Fuzzy IT2
1	1.0000 ±0.0000	1.0000 ±0.0000	1.0000 ±0.0000	0.9751 ±0.0012	1.0000 ±0.0000	1.0000 ±0.0000	1.0000 ±0.0000	0.9751 ±0.0012	1.0000 ±0.0000	1.0000 ±0.0000	1.0000 ±0.0000
2	1.0000 ±0.0000	1.0000 ±0.0000	1.0000 ±0.0000	0.9441 ±0.0016	1.0000 ±0.0000	1.0000 ±0.0000	1.0000 ±0.0000	0.9441 ±0.0016	1.0000 ±0.0000	1.0000 ±0.0000	1.0000 ±0.0000
3	1.0000 ±0.0000	1.0000 ±0.0000	1.0000 ±0.0000	0.9315 ±0.0027	1.0000 ±0.0000	1.0000 ±0.0000	1.0000 ±0.0000	0.9315 ±0.0027	1.0000 ±0.0000	1.0000 ±0.0000	1.0000 ±0.0000
4	0.9970 ±0.0004	0.9968 ±0.0005	1.0000 ±0.0000	0.9248 ±0.0021	1.0000 ±0.0000	1.0000 ±0.0000	1.0000 ±0.0000	0.9248 ±0.0021	0.9970 ±0.0005	0.9968 ±0.0005	1.0000 ±0.0000
5	0.9823 ±0.0014	0.9819 ±0.0015	0.9942 ±0.0008	0.9179 ±0.0023	1.0000 ±0.0000	1.0000 ±0.0000	1.0000 ±0.0000	0.9179 ±0.0023	0.9823 ±0.0014	0.9819 ±0.0015	0.9942 ±0.0008
6	0.9657 ±0.0023	0.9751 ±0.0016	0.9876 ±0.0011	0.9142 ±0.0026	1.0000 ±0.0000	1.0000 ±0.0000	1.0000 ±0.0000	0.9142 ±0.0026	0.9657 ±0.0023	0.9751 ±0.0016	0.9876 ±0.0011
7	0.9486 ±0.0018	0.9706 ±0.0016	0.9824 ±0.0013	0.9148 ±0.0030	1.0000 ±0.0000	0.9993 ±0.0003	0.9995 ±0.0002	0.9148 ±0.0030	0.9486 ±0.0018	0.9699 ±0.0018	0.9819 ±0.0014
8	0.9354 ±0.0015	0.9688 ±0.0019	0.9791 ±0.0016	0.9089 ±0.0026	0.9760 ±0.0016	0.9775 ±0.0011	0.9794 ±0.0009	0.9089 ±0.0026	0.9114 ±0.0027	0.9463 ±0.0026	0.9586 ±0.0018
9	0.9176 ±0.0029	0.9610 ±0.0024	0.9699 ±0.0023	0.9020 ±0.0025	0.9178 ±0.0022	0.9240 ±0.0028	0.9298 ±0.0023	0.9020 ±0.0025	0.8354 ±0.0033	0.8850 ±0.0035	0.8997 ±0.0031
10	0.8947 ±0.0029	0.9519 ±0.0023	0.9611 ±0.0021	0.8864 ±0.0026	0.8458 ±0.0035	0.8473 ±0.0023	0.8571 ±0.0030	0.8864 ±0.0026	0.7406 ±0.0044	0.7992 ±0.0024	0.8182 ±0.0028
11	0.8676 ±0.0030	0.9438 ±0.0017	0.9521 ±0.0015	0.8660 ±0.0044	0.7718 ±0.0044	0.7746 ±0.0036	0.7840 ±0.0030	0.8660 ±0.0044	0.6394 ±0.0055	0.7184 ±0.0032	0.7360 ±0.0029
12	0.8407 ±0.0032	0.9328 ±0.0019	0.9445 ±0.0018	0.8431 ±0.0052	0.7177 ±0.0063	0.7154 ±0.0048	0.7163 ±0.0055	0.8431 ±0.0052	0.5583 ±0.0043	0.6482 ±0.0047	0.6608 ±0.0051

Next, it can be observed that the overall RMSE measure was slightly lower for the IT2 fuzzy voter in majority of the considered noise levels. In other words, when both fuzzy voters were correct the IT2 fuzzy voter typically produced results that were closer to the reference correct output signals.

Fig. 16 also clearly demonstrates the fundamentally different arbitrating behavior of the averaging voting scheme. The first difference is that the averaging voter always offers an arbitrated result, since it does not maintain any means for detecting benign outputs. While this can be considered as a maximum availability, the results of the averaging voter feature substantially lower reliability and safety and higher RMSE for most of the considered noise levels. Fig. 16 further illustrates that there is a certain uncertainty threshold that marks the region of reasonable arbitration (in this test case around noise amplitude of 8). When the level of uncertainty (noise) increases above this threshold, simple averaging is the most reliable voting strategy.

Finally, in order to illustrate the difference in the voting behavior of both T1 and IT2 fuzzy voting schemes, the arbitrated outputs of 20 voting cycles are plotted in Fig. 17. Here, four different noise levels were considered. The figure also shows the manifestation of erroneous values in signal In_3 as the value of 20. It can be observed that the IT2 fuzzy voting scheme produces better results with the arbitrated signal closer to the reference correct signal when higher uncertainty levels are applied.

5.3. Parameter adjustment

The proposed transition from T1 fuzzy sets to the IT2 fuzzy sets of the voting system is controlled by the blurring parameter α. This parameter defines the amount of blur of the IT2 fuzzy sets and thus directly determines the size of the FOUs of individual fuzzy sets. In the previous section, an IT2 fuzzy voting system was examined, which was constructed with an arbitrarily selected blurring parameter of 1. Here the effect of adjusting the blurring parameter α is studied.

In this experiment four IT2 fuzzy voting systems are considered. They were all constructed from the T1 fuzzy voting system presented in Fig. 14, but with values of the blurring parameter: 0.25, 0.5, 0.75, and 1.0. All constructed voters have been

Table 3
Comparison of RMSE, RMSE on correct and RMSE on incorrect voting results, mean (bold italics font) and standard deviation (italics font).

Noise amplitude	RMSE				RMSE correct				RMSE incorrect			
	Average	Majority	Fuzzy T1	Fuzzy IT2	Average	Majority	Fuzzy T1	Fuzzy IT2	Average	Majority	Fuzzy T1	Fuzzy IT2
1	2.9911	0.3502	0.3502	0.3559	2.8080	0.3502	0.3502	0.3559	7.1101	0.0000	0.0000	0.0000
	±0.0349	±0.0023	±0.0023	±0.0023	±0.0400	±0.0023	±0.0023	±0.0023	±0.0035	±0.0000	±0.0000	±0.0000
2	3.0429	0.6993	0.7002	0.7179	2.5725	0.6993	0.7002	0.7179	7.3417	0.0000	0.0000	0.0000
	±0.0196	±0.0048	±0.0048	±0.0049	±0.0259	±0.0048	±0.0048	±0.0049	±0.0088	±0.0000	±0.0000	±0.0000
3	3.1251	1.0534	1.0695	1.0890	2.5035	1.0534	1.0695	1.0890	7.5730	0.0000	0.0000	0.0000
	±0.0326	±0.0035	±0.0031	±0.0030	±0.0202	±0.0035	±0.0031	±0.0030	±0.0116	±0.0000	±0.0000	±0.0000
4	3.2355	1.4487	1.4693	1.4605	2.5245	1.4487	1.4693	1.4605	7.7940	0.0000	0.0000	0.0000
	±0.0210	±0.0087	±0.0099	±0.0104	±0.0218	±0.0087	±0.0099	±0.0104	±0.0153	±0.0000	±0.0000	±0.0000
5	3.3972	2.0205	1.9468	1.8839	2.6096	2.0205	1.9468	1.8839	8.0272	0.0000	0.0000	0.0000
	±0.0273	±0.0124	±0.0105	±0.0105	±0.0200	±0.0124	±0.0105	±0.0105	±0.0085	±0.0000	±0.0000	±0.0000
6	3.5684	2.6229	2.5137	2.4006	2.7421	2.6229	2.5137	2.4006	8.2648	0.0000	0.0000	0.0000
	±0.0305	±0.0125	±0.0145	±0.0143	±0.0275	±0.0125	±0.0145	±0.0143	±0.0375	±0.0000	±0.0000	±0.0000
7	3.7299	3.2345	3.1074	2.9800	2.9134	3.2345	3.1025	2.9764	8.4959	0.0000	7.2453	7.1793
	±0.0388	±0.0141	±0.0118	±0.0112	±0.0215	±0.0141	±0.0107	±0.0104	±0.0378	±0.0000	±0.0564	±0.0404
8	3.9498	3.8379	3.7093	3.5844	3.0830	3.7013	3.5723	3.4551	8.7405	7.3324	7.4605	7.4344
	±0.0201	±0.0124	±0.0112	±0.0107	±0.0127	±0.0095	±0.0122	±0.0110	±0.0526	±0.0146	±0.0115	±0.0119
9	4.1578	4.4508	4.3099	4.2064	3.2649	3.9942	3.8778	3.7954	8.8486	7.6810	7.7318	7.7346
	±0.0254	±0.0199	±0.0193	±0.0190	±0.0204	±0.0185	±0.0217	±0.0186	±0.0265	±0.0150	±0.0236	±0.0295
10	4.3871	5.0138	4.8542	4.7820	3.3786	4.1134	3.9519	3.9336	8.9652	8.0377	8.0719	8.0750
	±0.0224	±0.0206	±0.0169	±0.0152	±0.0181	±0.0161	±0.0172	±0.0190	±0.0267	±0.0092	±0.0172	±0.0197
11	4.6313	5.5970	5.4120	5.3699	3.4742	4.1510	3.9903	4.0300	9.0585	8.4154	8.4792	8.4704
	±0.0379	±0.0231	±0.0291	±0.0268	±0.0149	±0.0229	±0.0235	±0.0228	±0.0453	±0.0177	±0.0238	±0.0272
12	4.8778	6.1503	5.9324	5.9629	3.5480	4.1236	3.9490	4.0172	9.1655	8.8887	8.9349	8.9879
	±0.0466	±0.0360	±0.0393	±0.0426	±0.0127	±0.0179	±0.0258	±0.0240	±0.0453	±0.0163	±0.0222	±0.0335

tested as described in Section 5.2. The improvements in availability, safety and reliability relative to the original T1 fuzzy voting system are summarized in Fig. 18.

Two observations can be made by studying Fig. 18. Firstly, it can be noted that voting behavior of both T1 and IT2 fuzzy voters is identical for small levels of noise, when all input signals maintain high similarity. The behavior of the IT2 fuzzy voter starts differentiating when the noise amplitude of the input signals reaches the region of *uncertain fuzzy agreement* of the voter. Secondly, it is important to observe that the IT2 fuzzy voting system might also cause performance deterioration as it is shown in Fig. 18 for safety and reliability. Hence, it is important to explore the design space of the voting system, in order to achieve robust voting system yielding performance improvement.

5.4. Scalability

The proposed IT2 fuzzy voting system was presented on an example of a TMR system. However, the developed voting system is easily extendable for systems with higher dimensionality. Additional experiments have been performed to validate the scalability of the benefits of incorporating IT2 fuzzy logic into the design of the fuzzy voting system.

The experimental harness presented in Fig. 13 was extended for the case of 4-way and 5-way voter. All 4-way and 5-way average, majority, T1 fuzzy and IT2 fuzzy voters have been implemented. The testing methodology remained identical as described in Section 5.2. The architectures of each T1 and IT2 fuzzy logic system computing the weight for each input were also identical as shown in Fig. 14. The fuzzy rule base was extended to account for 3 and 4 differences to the other input signal for the 4-way and the 5-way voters, respectively.

The availability, safety, reliability and the RMSE measures were computed for both 4-way and 5-way voters over 10,000 voting cycles. The results are presented in Figs. 19 and 20. The presented experimental results demonstrated that the advantageous properties of the IT2 fuzzy voter are preserved when the dimensionality of the systems increases. In both Figs. 19 and 20 the 4-way and 5-way IT2 fuzzy voters provide increased availability, safety and reliability together with reduced RMSE of the arbitrated results, when compared to the majority and the T1 fuzzy voters. In addition, Table 4 summarizes

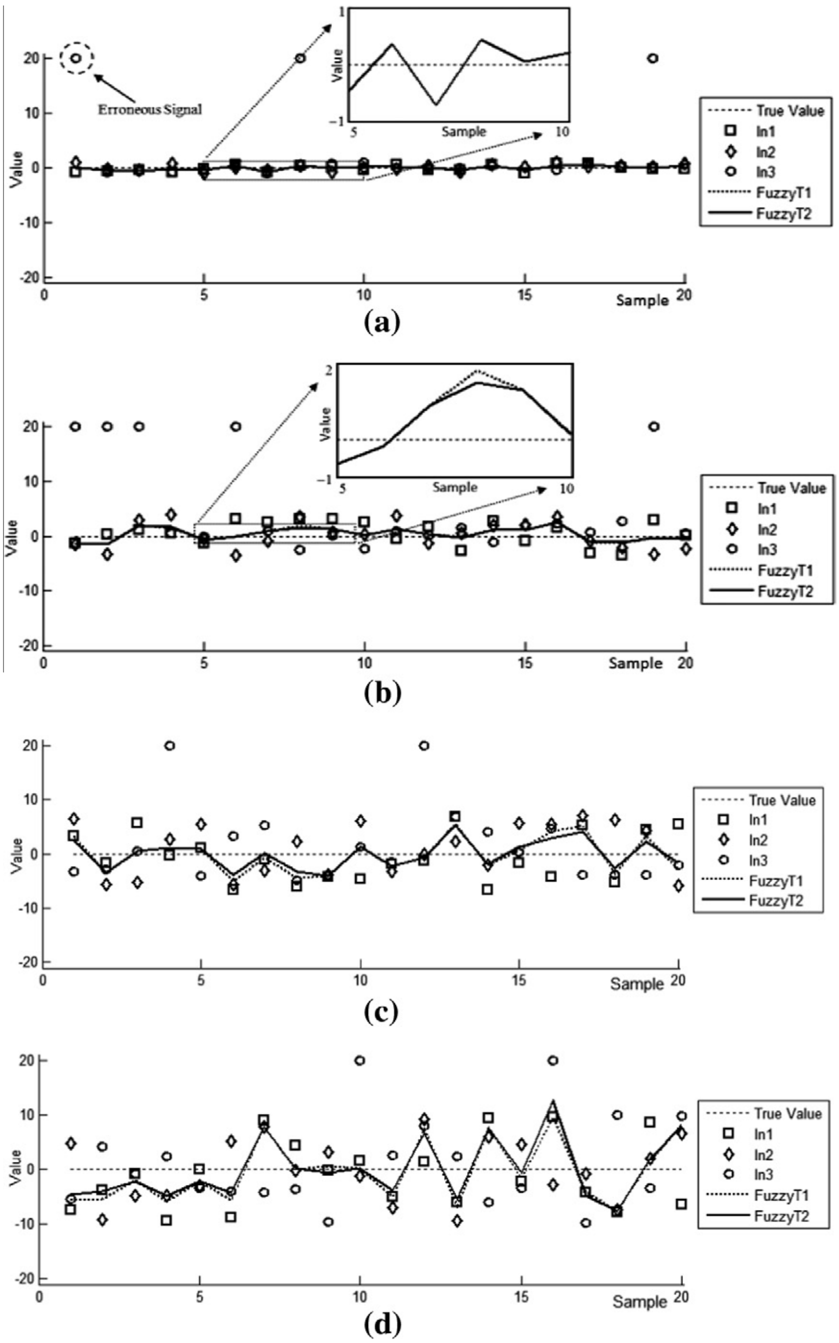

Fig. 17. Recorded arbitrated outputs for different input signals with noise amplitude of 1 (a), 4 (b), 7 (c) and 10 (d).

the relative average performance improvements of the T1 and IT2 fuzzy voters with respect to the majority voter for all the considered levels of uncertainty. From the table it can be concluded that both voting schemes provided significant performance improvement relative to the majority voter and that the IT2 fuzzy voter consistently outperformed the T1 fuzzy voting design. The relationship was preserved despite the dimensionality of the input.

Overall, it can be concluded that when an excessive amount of uncertainty is present, the best that a voting system can do is to average the input signals together. However, when reasonable noise amplitudes are present, the inexact majority and the fuzzy voting schemes provide substantially improved voting performance. The IT2 fuzzy voting scheme offers improved availability, reliability and safety in a wide range of uncertainty amplitudes. The performance of the T1 and IT2 fuzzy voters and the inexact majority voters are nearly identical within the small amount of uncertainty.

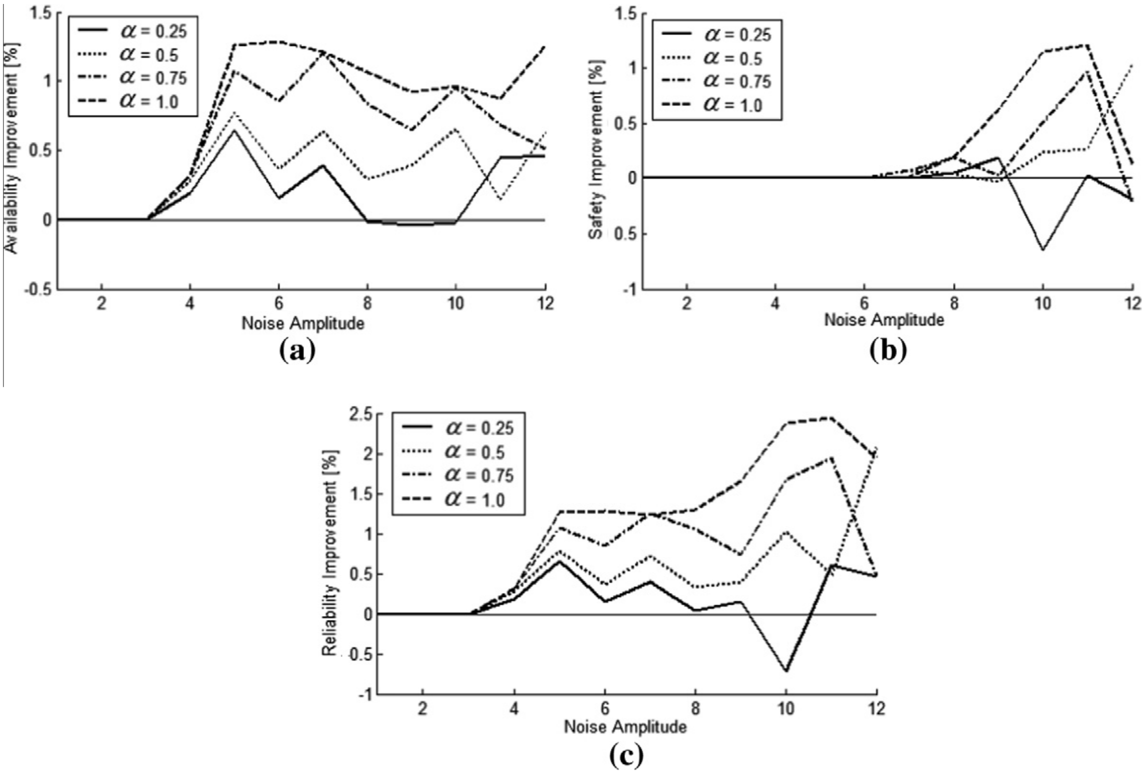

Fig. 18. The relative improvements in availability (a), safety (b) and reliability (c) for IT2 fuzzy systems constructed with $\alpha = 0.25, 0.5, 0.75, 1$.

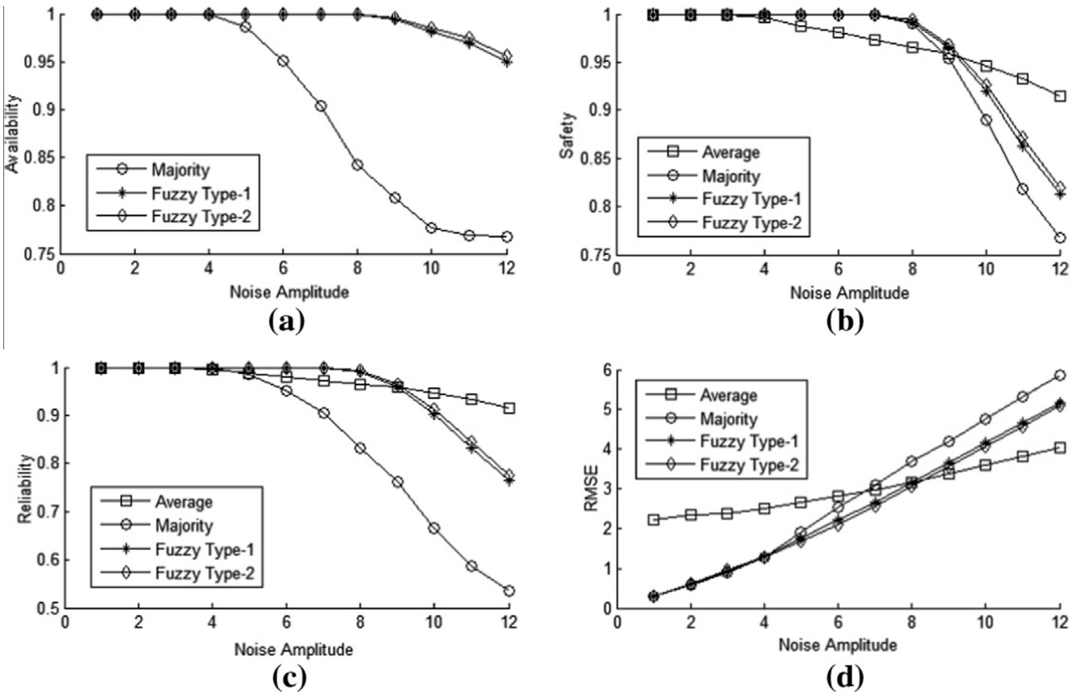

Fig. 19. The availability (a), safety (b), reliability (c) and RMSE (d) measures for the 4-way voters.

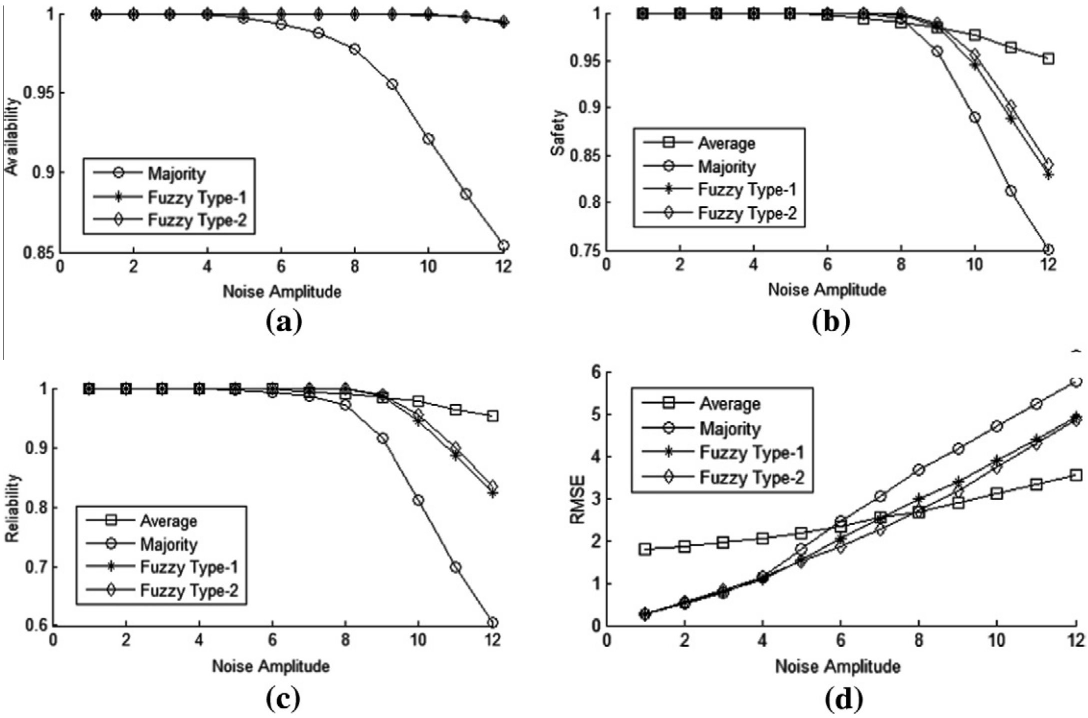

Fig. 20. The availability (a), safety (b), reliability (c) and RMSE (d) measures for the 5-way voters.

Table 4
Average relative performance improvement of the T1 and IT2 fuzzy voters over the majority voter.

Number of inputs	FLS type	Availability (%)	Safety (%)	Reliability (%)	RMSE (%)
3	Type-1	3.14	0.08	4.11	2.10
	Interval Type-2	3.94	0.36	5.34	3.20
4	Type-1	11.22	1.35	15.12	7.92
	Interval Type-2	11.40	1.62	15.72	9.15
5	Type-1	3.88	2.43	7.68	10.90
	Interval Type-2	3.9	2.81	8.13	12.77

6. Conclusion

This paper presented a novel design of an Interval Type-2 fuzzy voting scheme. The architecture of the IT2 fuzzy voter was obtained by asymmetrically blurring the membership functions of the respective T1 fuzzy voting system. This specific asymmetric blurring methodology was presented to accentuate the amalgamating capability for cases of definite agreement and uncertain agreement among inputs of the fuzzy voter and to preserve the outlier detection capability for definitely disagreeing input signals.

The performance of the proposed IT2 fuzzy voter was tested on a refined experimental harness. The testing system allows for modeling of various distributions of both noise and errors in the input signal. The performance of the IT2 fuzzy voting system was compared to the T1 fuzzy voter, the averaging voter and the inexact majority voter.

The obtained results demonstrated that the IT2 fuzzy voter increases the availability of the voting system, while maintaining high safety and reliability. In addition, the RMSE measure of the voter outputs was lower for most of the considered noise levels than of the other voting systems. Further experiments demonstrated that this advantageous voting behavior is preserved for higher-dimensional systems. Overall, the experimental results showed that an appropriate management of the amount of blur in the IT2 fuzzy sets can lead to robust voting behavior. In addition, it was concluded that when an excessive amount of uncertainty is present, the best that a voting system can do is to average the input signals together. When reasonable noise amplitudes are present, the inexact majority and the fuzzy voting schemes provide substantially improved voting performance.

References

[1] M. Beglarbegian, W. Melek, J.M. Mendel, On the robustness of type-1 and interval type-2 fuzzy logic systems in modeling, Information Sciences 181 (7) (2011) 1325–1347.
[2] S. Bennett, G. Latif-Shabgahi, Evaluation of the performance of voting algorithms used in fault tolerant control systems, in: Proc. 14th World Congress of International Federation of Automatic Control, vol. Q, Beijing, China, 1999, pp. 525–530.
[3] S.-M. Chen, L.-W. Lee, Fuzzy multiple attributes group decision-making based on the ranking values and the arithmetic operations of interval type-2 fuzzy sets, Expert Systems with Applications 37 (2010) 824–833.
[4] B.-I. Choi, F.Ch.-H. Rhee, Interval type-2 fuzzy membership function generation methods for pattern recognition, Information Sciences 179 (13) (2009) 2102–2122.
[5] S. Coupland, R. John, An investigation into alternative methods for the defuzzification of an interval type-2 fuzzy set, in: Proc. IEEE International Conference on Fuzzy Systems, Vancouver, Canada, pp. 1425–1432, 2006.
[6] S. Coupland, R. John, Geometric type-1 and type-2 fuzzy logic systems, IEEE Transactions on Fuzzy Systems 15 (1) (2007) 3–15.
[7] G. Deschrijver, Arithmetic operators in interval-valued fuzzy set theory, Information Sciences 177 (14) (2007) 2906–2924.
[8] S. Greenfield, R John, S. Coupland, A novel sampling method for type-2 defuzzification, in: Proc. UKCI 06, pp. 120–127, 2005.
[9] S. Greenfield, F. Chiclana, S. Coupland, R. John, The collapsing method of defuzzification for discretised interval type-2 fuzzy sets, Information Sciences (179) (2009) 2055–2069.
[10] H.A. Hagras, A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots, IEEE Transactions on Fuzzy Systems 12 (4) (2004) 524–539.
[11] H. Hamrawi, S. Coupland, R. John, A novel alpha-cut representation for type-2 fuzzy sets, in: Proc. of IEEE World Congress on Computational Intelligence, Barcelona, Spain, pp. 351–358, 2010.
[12] E. Herrera-Viedma, S. Alonso, F. Chiclana, F. Herrera, A consensus model for group decision making with incomplete fuzzy preference relations, IEEE Transactions on Fuzzy Systems 15 (5) (2007) 863–877.
[13] H. Ishibuchi, T. Nakashima, T. Morisawa, Voting in fuzzy rule-based systems for pattern classification problems, Fuzzy Sets and Systems 103 (1999) 223–238.
[14] N.N. Karnik, J.M. Mendel, Type-2 fuzzy logic systems, IEEE Transactions on Fuzzy Systems 7 (6) (1999) 643–658.
[15] N.N. Karnik, J.M. Mendel, Centroid of a type-2 fuzzy set, Information Sciences 132 (2001) 195–220.
[16] L.I. Kuncheva, Fuzzy versus nonfuzzy in combining classifiers designed by boosting, IEEE Transactions on Fuzzy Systems 11 (6) (2003) 729–741.
[17] G. Latif-Shabgahi, J.M. Bass, S. Bennet, Efficient implementation of inexact majority and median voters, Electronics Letters 36 (15) (2000).
[18] G. Latif-Shabgahi, J.M. Bass, S. Bennett, A taxonomy for software voting algorithms used in safety-critical systems, IEEE Transaction on Reliability 53 (3) (2004) 319–328.
[19] G. Latif-Shabgahi, A.J. Hirst, A fuzzy voting scheme for hardware and software fault tolerant systems, Fuzzy Sets and Systems 150 (3) (2005) 579–598.
[20] O. Linda, M. Manic, Comparative analysis of type-1 and type-2 fuzzy control in context of learning behaviors for mobile robotics, in: Proc. the 36th Annual Conference of the IEEE Industrial Electronics Society, 2010.
[21] F. Liu, An efficient centroid type-reduction strategy for general type-2 fuzzy logic system, Information Sciences 178 (2008) 2224–2236.
[22] M. Manic, D. Frincke, Towards the fault tolerant software: fuzzy extension of crisp voters, in: Proc. of the 27th Annual Conference of IEEE, pp. 84–89, 2001.
[23] R. Martinez, O. Castillo, L.T. Aguilar, Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms, Information Sciences 179 (13) (2009) 2158–2174.
[24] K. Marzullo, Tolerating failures of continuous-valued sensors, ACM Transactions on Computer Systems 8 (1990) 284–304.
[25] F. Mata, L. Martinez, E. Herrera-Viedma, An adaptive consensus support model for group decision-making problems in a multigranular fuzzy linguistic context, IEEE Transactions on Fuzzy Systems 17 (2) (2009) 279–290.
[26] J.M. Mendel, Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, Prentice-Hall, Upper Saddle River, NJ, 2001.
[27] J.M. Mendel, R. John, F. Liu, Interval type-2 fuzzy logic systems made simple, IEEE Transactions on Fuzzy Systems 14 (6) (2006).
[28] J.M. Mendel, F. Liu, D. Zhai, α-Plane representation for type-2 fuzzy sets: theory and applications, IEEE Transactions on Fuzzy Systems 17 (5) (2009) 1189–1207.
[29] M. Nie, W.W. Tan, Towards and efficient type-reduction method for interval type-2 fuzzy logic system, in: Proc. of FUZZ-IEEE, Hong Kong, pp. 1425–1432, 2008.
[30] B. Parhami, Voting algorithms, IEEE Transactions on Reliability 43 (4) (1994) 617–629.
[31] S. Tosunoglu, Fault tolerance for modular robots, Proceedings of IECON 3 (1993) 1910–1914.
[32] J.A. Sanz, A. Fernandez, H. Bustince, F. Herrera, Improving the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets and genetic amplitude tuning, Information Sciences 180 (19) (2010) 3674–3685.
[33] I.B. Turksen, Non-specificity and interval-valued fuzzy sets, Information Sciences 80 (1) (1996) 87–100.
[34] Ch. Wagner, H. Hagras, A genetic algorithm based architecture for evolving type-2 fuzzy logic controllers for real world autonomous mobile robots, in: Proc. FUZZ-IEEE, London, pp. 1–6, 2007.
[35] Ch. Wagner, H. Hagras, Toward general type-2 fuzzy logic systems based on zSlices, IEEE Transaction of Fuzzy Systems 18 (4) (2010) 637–660.
[36] D. Wu, W.W. Tan, Genetic learning and performance evaluation of interval type-2 fuzzy logic controllers, Engineering Applications of Artificial Intelligence 19 (8) (2006) 829–841.
[37] D. Wu, J.M. Mendel, Uncertainty measures for interval type-2 fuzzy sets, Information Sciences 177 (23) (2007) 5378–5393.
[38] D. Wu, J.M. Mendel, Enhanced Karnik–Mendel algorithm, IEEE Transactions on Fuzzy Systems 17 (4) (2009) 923–934.
[39] S.-Y. Yu, N. Saxena, E.J. McCluskey, ACS implementation of a robotic control algorithm with fault tolerant capabilities, in: Proc. IEEE Symp. Field-Programmable Custom Computing Machines, Napa Valley, California, pp. 175–184, 2000.
[40] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning – I, Information Sciences 8 (1975) 199–249.
[41] D. Zhai, J.M. Mendel, Uncertainty measures for general type-2 fuzzy sets, Information Sciences 181 (3) (2011) 503–518.