Type-2 Radial Basis Function Neural Networks Mandami System to Evaluate Quality Features

Pascual Noradino Montes Dorantes (pascualresearch@gmail.com)
Instituto Tecnológico de Ciudad Victoria
Gerardo Maximiliano Méndez
Tecnológico Nacional de México/ Instituto Tecnológico de Nuevo León, Av. Eloy Cavazos 2001, Guadalupe Nuevo León, México, Posgrado en Ingeniería Mecatrónica
Marco Aurelio Jiménez Gómez
Tecnológico Nacional de México/ Instituto Tecnológico de Ciudad Victoria/División de estudios de posgrado e investigación. Boulevard Emilio Portes Gil #1301 Pte. A.P. 175, C.P. 87010, Ciudad Victoria, Tamaulipas, México
Adriana Mexicano Santoyo
Tecnológico Nacional de México/ Instituto Tecnológico de Saltillo. División de estudios de posgrado e investigación Blvd. V. Carranza No. 2400, Col. Tecnológico, CP. 25280 Saltillo, Coah. México

---

Research Article

Keywords: Radial basis function network, RBFNN, interval type-2, IT-2, image processing, quality assurance

Posted Date: September 3rd, 2021

DOI: https://doi.org/10.21203/rs.3.rs-862636/v1

Type-2 Radial basis function neural networks Mandami System to Evaluate Quality Features

Pascual Noradino Montes Dorantes [a,c] *, Gerardo Maximiliano Méndez[b], Marco Aurelio Jiménez Gómez[d], Adriana Mexicano Santoyo[c]

a. Universidad Autónoma del Noreste. División de estudios de posgrado e investigación. Blvd. José Musa de León y General Medardo de la Peña S/N, Col. Los Pinos, C.P. 25100, Saltillo, Coahuila, México. https://orcid.org/0000-0001-8804-9623
b. Tecnológico Nacional de México/ Instituto Tecnológico de Nuevo León, Av. Eloy Cavazos 2001, Guadalupe Nuevo León, México, Posgrado en Ingeniería Mecatrónica. Ph: (52) 8181 57 05 00 Ext. 140. gmm_paper@yahoo.com.mx. https://orcid.org/0000-0002-7377-4141
c. Tecnológico Nacional de México/ Instituto Tecnológico de Saltillo. División de estudios de posgrado e investigación Blvd. V. Carranza No. 2400, Col. Tecnológico, CP. 25280 Saltillo, Coah. México Tel: (844) 438 95 39 m_jimenez81@yahoo.com.mx. https://orcid.org/0000-0002-5281-3615
d. Tecnológico Nacional de México/ Instituto Tecnológico de Ciudad Victoria/División de estudios de posgrado e investigación. Boulevard Emilio Portes Gil #1301 Pte. A.P. 175, C.P. 87010, Ciudad Victoria, Tamaulipas, México. Ph. (52) 834 153 2000 ext. 306. mexicanoa@gmail.com. https://orcid.org/0000-0002-8045-7558
e.

* Corresponding author.
E-mail address: pascualresearch@gmail.com; cualmontes@hotmail.com (P.N. Montes).

Abstract

This paper presents type-1 and type-2 radial basis function networks to evaluate quality features. The proposed methodology fuses the central composite design and the radial basis function neural networks in type-1 or interval type-2 model to generate a network that evaluates quality features in an industrial image processing. The advantages of this proposal include that training is not required to get an accurate result and that the generation of the fuzzy rule base using central composite design method and statistical indicators is simplified. Another advantage is the excellent results obtained with the proposal. Experimentation shows an error reduction of 90% when the interval type 2 Mandami Radial basis function neural network compared against its type-1 counterpart using the Gaussian membership functions onto a radial basis function network.

Keywords: Radial basis function network, RBFNN, interval type-2, IT-2, image processing, quality assurance.

1. Introduction

Interval type-2 (IT2) systems arise as an alternative to manage uncertainties present in all industrial process. On the other side, type-1 (T1) models cannot manage uncertainties as is mentioned by Mendel [1] such as T1 singleton fuzzy logic models or their equivalent radial basis function neural networks (RBFNN); T1 models also require several cycles, epochs or iterations of training and adjustment to get an acceptable results or an adequate level of precision such as is presented by [2] in which 1300 epochs of training has been required to get an adequate result. But nevertheless, the main problem to obtain a more precise output relies on the modeling of the system as is mentioned by [3], most researchers base their modeling by intuition and as a thumb rule. Also, specific criteria for how many rules are needed to modeling a system does not exist.

The literature shows a couple of techniques to model intelligent systems [4-11] but all of them are applied toT1 models. Only a pair of proposals is found that uses the IT2 [2, 11]. In [4-5] has been proposed a model that uses linguistic labels as fuzzy sets, but in [5] the center label is missing to create the universe of discourse

(UOD) and with them the fuzzy rule base. The [4] model is restricted to odd input variables and [5] is restricted to even input variables.

In [6] has been developed a method that uses control charts to model the rule base, but it is restricted to: the addition of a genetic algorithm to generate the membership functions (MF), the support or the width of the fuzzy sets is non-uniform, and finally that a combination of trapezoidal and triangular MF's is produced to get the UOD. In [7] a similar method to [6] has been used but it only uses triangular MF's and it also presents the presence of blank spaces in the UOD. In [8] five linguistic labels have been used which are restricted to a constant rule base that is inflexible and unadaptable with blank spaces in the UOD.

The statistical properties of the Gaussian distribution are used in [3] and [9] to model the fuzzy rule base. The central composite design (CCD) of the design of experiments is used in [10-11] to get a simplified and compact rule base to assemble the UOD. In [10] a T1 RBFN that does need training has been proposed. In [11] the technique of CCD is used to model an IT2 rule base that considers uncertainties.

Some theoretical proposals exist in IT2 RBFNN for the network modelling and assemble. A framework is presented in [2] to model a IT2 network for the first time, but it is mentioned the necessity of expert knowledge to assemble the IT2. This is not the only one solution, in [1] is stated that "Rules may be provided by experts or can be extracted from numerical data", such is demonstrated in [3, 4, 9-11]. A general IT2 network is presented in [12], but it is restricted to only one type of membership function, as a Gaussian function. A recurrent self-evolving RBFNN is presented in [13] to adapt the neurons in the hidden layer of the network in a nonlinear dynamical system. Ngo in [14] has proposed a model to convert the IT2 Mandami model into an IT2 Takagi-Sugeno model via genetic algorithms; Baklouti et al in [15] have applied an IT2 RBFNN to type TSK in order to evaluate time series.

In the classic approach of the RBFNN can only be used two types of membership functions. In [16] only the logistic function, see Fig. 1, or the Gaussian function, see Fig. 2, that are defined as receptive fields or fuzzyfiers (which were presented by [3]).

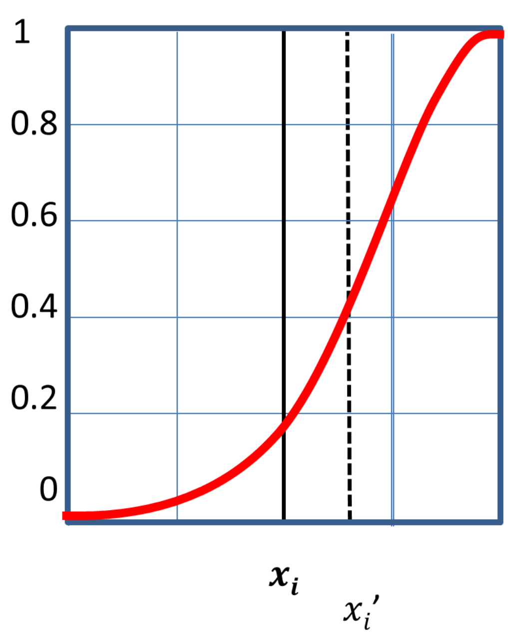

Fig. 1. Logistic function.

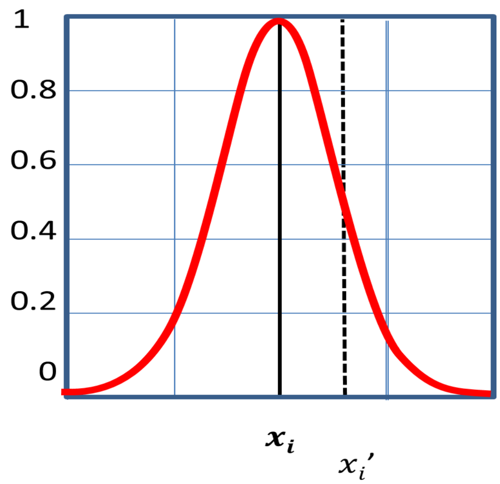

Fig. 2. Gaussian function.

2. *Theoretical foundations*

2.1 T1 Radial basis function neural networks

This kind of neural network is a type of interpolation focused on getting approximations based on a distance between a pattern and a sample [16] and is graphically depicted on Fig.3. The RBFNN is similar to the K-Nearest Neighbor's (KNN) classifier.

The T1 RBFNN operates with assigned weights to the inputs in which each weight is placed in a multidimensional vector given by (1) and is based on a Gaussian distribution function (2) or by a logistic distribution function (3). This Gaussian distribution function is equivalent to a membership function in fuzzy models. The output of the RBFNN model could be calculated in many forms. Thought

$$w_i = R_i(x) = R_i\left(\left\|x - u_x\right\| \frac{u_i}{\sigma_x}\right) \qquad \forall\, x, u \in U \tag{1}$$

where: u_x is the mean of the in the radial basis function, x is the input of the function and σ_x is the spread of the function.

$$R(x) = e^{-\frac{\|x-u_x\|^2}{2\sigma_x{}^2}} \qquad \forall\, x, u \in U \tag{2}$$

$$R(x) = \frac{1}{1+\frac{e^{[\|x-u_i\|]^2}}{\sigma_i^2}} \forall\, x, u \in U \tag{3}$$

$$d(x) = \sum_{i=1}^{H} c_i w_i \;\forall\, c, R \in x \tag{4}$$

$$d(x) = \sum_{i=1}^{H} c_i R_i \;\forall\, c, R \in x \tag{5}$$

$$d(x) = \frac{\sum_{i=1}^{H} c_i w_i}{\sum_{i=1}^{H} w_i} \tag{6}$$

$$d(x) = \frac{c_i R_i(x)}{R_i(x)} \quad \forall\, c, R \in x \tag{7}$$

where: c_i is the output in the dataset for the prediction and the final approximation is obtained by (5, 6 or 7) which can be reinterpreted as a center of gravity deffuzifier in fuzzy models (8).

$$y_c(x) = \frac{\sum_1^n y_i \mu_B(y_i)}{\sum_1^n \mu_B(y_i)} \tag{8}$$

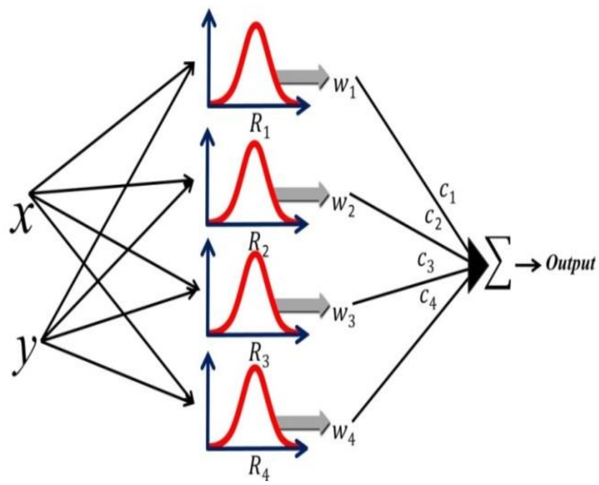

Fig. 3. RBFNN of T1 topology, case with two inputs.

2.2 IT2 Fuzzy logic system

The IT2 arises from the union of two T1 fuzzy sets represented by (9) to generate an interval. An IT2 fuzzy set is given by (10) and it is characterized by $\tilde{A}$ with a membership function $\mu_{\tilde{A}}(x,u)$,

$$A = \{(x, \mu_A(x)) / \forall x \in X\} \tag{9}$$

where: A represents the set and μ_A is the grade of membership of some *x'* in *A*.

$$\tilde{A} = \{((x,u), \mu_{\tilde{A}}(x,u)) \,|\, \forall x \in X, \forall u \in J_x \subseteq [0,1]\} \tag{10}$$

The classic approach for the IT2 models is the Gaussian fuzzifier defined on (11), showing a difference against the T1 membership function that presents two means.

$$\mu_{\tilde{A}_k^i}(x_k) = \tilde{F}_k^i = exp\left[-\frac{1}{2}\left[\frac{x_k - m_k^i}{\sigma_k^i}\right]^2\right] \tag{11}$$

where $m_k^i \in [m_{k1}^i, m_{k2}^i]$ is the uncertain mean, $k = 1,2,\ldots,p$ (p is the number of inputs) and i = 1,2,..M (The number of M rules), and σ_k^i is the standard deviation.

2.3 IT2 Radial basis function networks

IT2 RBFNN has been presented for the first time in 2015 by [2]. And since then it has evolved into new models such as the ones presented by Mandami [2], Takagi-Sugeno [14] and Takagi-Sugeno-Kang model [15].

The basis of this model is the extension of the initial model showed by [16] to obtain the left and right functions (Fig. 4). Basically the similitudes of the T1 fuzzy model are equivalent to the T1 RBFNN. These are well known since 1993 in [17-19].

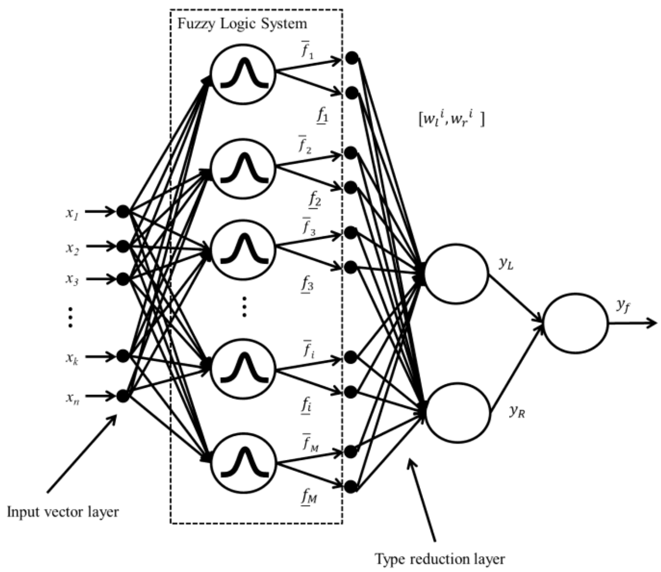

Fig. 4 IT2 RBFNN topology, from [2].

The IT2 RBFNN algorithm is generated with the adjustment of T1 RBFNN. First of all (2) is replaced with (11). Secondly (4-7) are converted to (12-15) to get the lower output and similarly (16-19) to get the upper output,

$$\underline{d(x)} = \sum_{i=1}^{H} \underline{c_i R_i}(x) \ \forall\, c, R \ \in x \tag{12}$$

$$\underline{d(x)} = \sum_{i=1}^{H} \underline{c_i w_i} \ \forall\, c, R \ \in x \tag{13}$$

$$\underline{d(x)} = \frac{\sum_{i=1}^{H} \underline{c_i w_i}}{\sum_{i=1}^{H} \underline{w_i}} \tag{14}$$

$$\underline{d(x)} = \frac{\underline{c_i R_i(x)}}{\underline{R_i}(x)} = \qquad \forall\, c, R \ \in x \tag{15}$$

$$\overline{d(x)} = \sum_{i=1}^{H} \overline{c}_i \, \overline{R}_i(x) \ \forall\, c, R \ \in x \tag{16}$$

$$\overline{d(x)} = \sum_{i=1}^{H} \overline{c}_i \overline{w}_i \ \forall\, c, R \ \in x \tag{17}$$

$$\overline{d(x)} = \frac{\sum_{i=1}^{H} \overline{c}_i \overline{w_i}}{\sum_{i=1}^{H} \overline{w_i}} \tag{18}$$

$$\overline{d(x)} = \frac{\overline{c}_i \overline{R}_i(x)}{\overline{R}_i(x)} = \qquad \forall\, c, R\ \in x \tag{19}$$

where *ci* is the output in the database for the prediction and the final approximation is obtained by (20) which can be reinterpreted as (21),

$$d(x) = \frac{\overline{d(x)} + \underline{d(x)}}{2} \tag{20}$$

$$d(x) = \frac{\left(\frac{\overline{c}_i \overline{R}_i(x)}{\overline{R}_i(x)}\right) + \left(\frac{\underline{c}_i \underline{R}_i(x)}{\underline{R}_i(x)}\right)}{2} \tag{21}$$

A. Central Composite design

The factorial design or CCD is a technique used to analyze the factors and their possible correlation or not a correlation. These factors present limits to define the universe and their levels are called low and high. In order to establish the model, a combinatorial via permutations is needed to define a pattern where one the variables must change meanwhile the rest remain constant. The simplest model is called 2^k (Fig. 5).

In Fig. 5, each sing at a corner represents different levels of variables. For *(1)* both variables are in low level, for *(a)* the first variable is on high level and the second one is on the low level, for *(b)* the first variable is on the in low level and the second on high level, and for *(ab)* both variables are on high level.

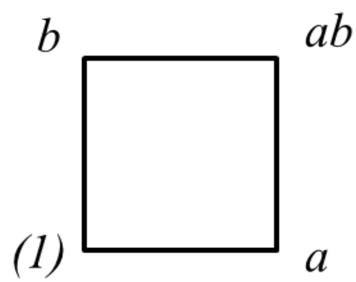

Fig. 5. 2^k CCD model.

3. Revisited equivalence of the T1 FLS and T1 RBFNN

In [2] a series of restrictions have been established to this equivalence as follows:

- The number of perceptive fields in the hidden layer is equal to the number of fuzzy rules.

First, a definition for the receptive field is required.

Definition 1: a receptive field is a neuron in the network that represents a mathematical operation such as (2) or (3).

But this restriction needs to be enhanced and Fig. 3 needs to be redrawn since the receptive field on this figure only represents a part of the fuzzy rule. e.g. in T1 model a fuzzy rule is defined by (22), and for every variable in the RBFNN it is required a receptive field represented mathematically by (2 or 3), then the number of receptive fields (*rf*) required is equal to the number of variables (*V*) multiplied by the number of rules (r) given by (23). The enhanced graphical representation is depicted on Fig. 6,

$$Rule\ i{:}\ IF\ x1\ is\ a\ and\ x2\ is\ b\ then\ y\ is\ G \tag{22}$$

where: $x1$, $x2$ are the inputs and G is the output of the rule.

$$rf = V * r \tag{23}$$

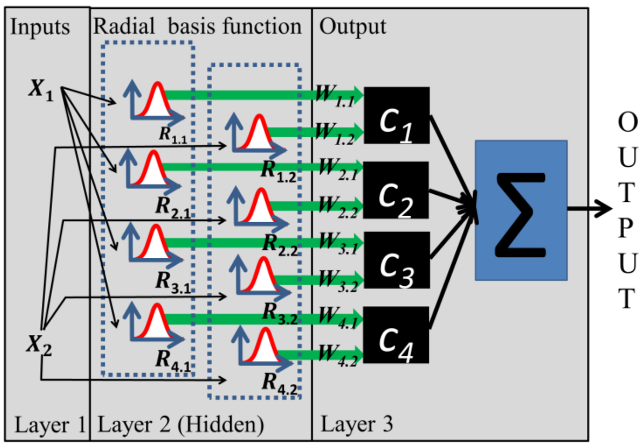

Fig. 6. RBFNN of T1 topology, two input case expanded.

The IT2 RBFNN topology of [2] shown in Fig. 4 needs to be redrawn (Fig 8) because Fig. 4 has additional receptive fields, this condition requires additional calculus and training to tuning precise output, also require additional receptive fields to achieve a precise output. Then, the quantity of receptive fields is given by (24), and the topology is reorganized with the addition of the new receptive fields,

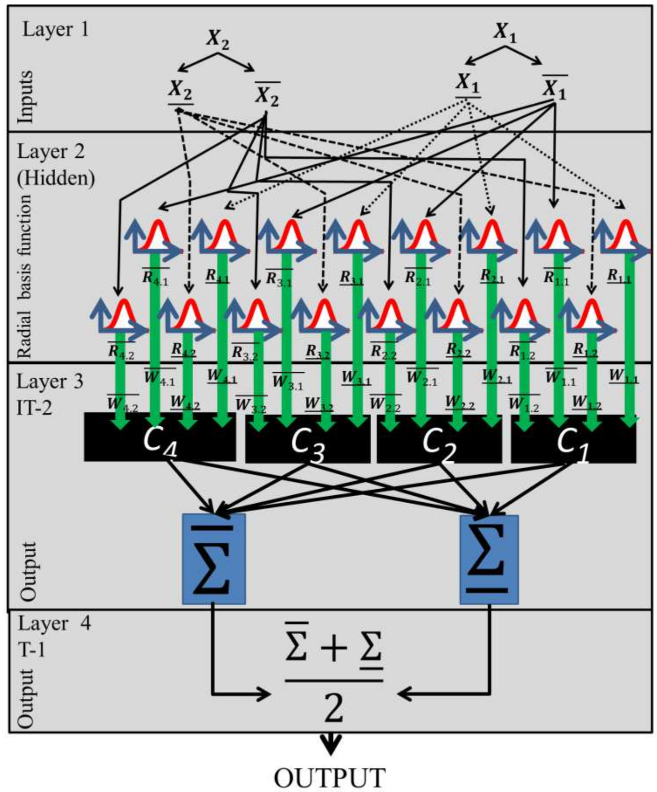

Fig. 7. IT2 RBFNN topology, two input case expanded.

$$rf = 2(V * r) \tag{24}$$

where: rf are the receptive fields, V is the number of variables and r is the number of rules are.

- The membership function within each rule is chosen as Gaussian.

But this fact requires testing because in [16] its mentioned that the logistic function is an alternative model of RBF but, that form of MF's need to be tested. Also, a definition of membership function is required in the neural network.

Definition 2: The membership function for neural networks is a function that defines the weight for the input variable.

- The T-norm operator used to compute each rule's firing strength is the multiplication.

The T-norm has been defiendd in [20] as the intersection of two sets. But, the intersection in the case of T1 RBFNN does not use in the receptive fields as occurs with the fuzzy rules before fuzzification in the FLS.

- Both the T1 RBFNN and the Fuzzy Inference System being considered here use the same defuzzification method using the center of gravity deffuzifier given by (8).

4. ***Proposal***

B. Assemble and calculation of IT2 RBFNN parameters based on the CCD

From CCD and [11] the IT2 CCD is obtained and used to generate the fuzzy rule base that serves as inputs for the IT2 RBFNN.

The initial calculi of the rule base or the inputs for the IT2 RBFNN are given by (25) as being proposed in [11]. These are obtained from the process control specifications and their equivalences to the CCD 2^k model are presented in Table 1,

Table 1
Equivalences of CCD states and quality control limits.

CCD Symbolic Representation	**Treatment**	**Quality control limits**
ab=1	A $_{low}$, B $_{low}$	$LCLa$ $LCLb$
Ab=a	A $_{high}$, B $_{low}$	$LCLa$ $UCLB$
aB =b	A $_{low}$, B $_{High}$	$LCLA$ $LCLb$
AB=ab	A $_{high}$, B $_{High}$	$UCLA$ $UCLB$

$$2^k = \begin{bmatrix} LCLa & LCLb \\ LCLa & UCLB \\ UCLA & LCLb \\ UCLA & UCLB \end{bmatrix} \tag{25}$$

where: 2^k is the matrix that conforms the CCD model and every pair in the matrix is formed by the possible combinations of the lower and upper limits of control for the input variables.

From (25) the inputs are obtained for the receptive fields on the IT2 RBFNN. The input for the first receptive field in the T1 RBFNN is *LCLa* and for calculating the receptive fields for the first variable in IT2 RBFNN additional calculations are needed. Firstly, it is needed the spread of the data specifications and it is given by (26),

$$\sigma_{x_i} = \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}_i}{n} \right) \tag{26}$$

With the matrix presented in (25) and σ_{x_i} from (26) the IT2 matrix could be calculated. The lower limit of interval represented by $\underline{L}$ is given by (27) and the upper interval limit $\overline{R}$ is given by (28), and their respective solutions $\overline{y}$ upper and $\underline{y}$ lower are obtained by interpolation, which are given by (29 and 30).

The equation (25) is converted into equation (31) to get data in the receptive field in the IT2 RBFNN,

$$\underline{L} = x_i \text{-} \sigma_{x_i} \tag{27}$$

$$\overline{R} = x_i + \sigma_{x_i} \tag{28}$$

$$\underline{y} = y_i \text{-} \sigma_{y_i} \tag{29}$$

$$\overline{y} = y_i + \sigma_{y_i} \tag{30}$$

$$Rf = \begin{bmatrix} \underline{a} & \overline{a} & \underline{b} & \overline{b} & \underline{y_1} & \overline{y_1} \\ \underline{a} & \overline{a} & \underline{B} & \overline{B} & \underline{y_b} & \overline{y_b} \\ \underline{A} & \overline{A} & \underline{b} & \overline{b} & \underline{y_a} & \overline{y_a} \\ \underline{A} & \overline{A} & \underline{B} & \overline{B} & \underline{y_{ab}} & \overline{y_{ab}} \end{bmatrix} \tag{31}$$

For i=1, a, b, ab.

where: Rf represents the receptive field universe, $\underline{a}$ represents the lower limit of the IT2 for the variable a and it is the first receptive field in the RBFNN. The same case is for the rest of the elements in the matrix Rf. The interval for the output response is obtained from $\underline{Y_i}, \overline{Y}_i$. Those values represent the lower and upper output or response for a specific receptive field of the inputs.

C. Enhancement and changes to the classic IT2 RBFNN

First, the RBF was changed from logistic to a Gaussian function form (3).

Second, the use of the CCD IT2 from [11] is adapted to model the IT2 RBFNN using (25-31).

5. *Results*

The results of this proposal are organized as follows. First it is presented the test for the logistic function as radial basis or receptive field in IT2 RBFNN without type reduction (Fig. 8) and IT2 RBFNN with type reduction (Fig. 9). Secondly, it is presented the test for the Gaussian RBF as receptive field in IT2 RBFNN without type reduction (Fig. 10) and IT2 RBFNN whit type reduction (Fig. 11). All experiments were made with a non-iterative RBFNN model; this is a network that does need training due to the application and the modeling basis.

To calculate the accuracy and the enhancement of the proposal is used the mean square error to document the variations given by (33),

$$MSE = \frac{\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2}{n}$$

where: $\hat{Y}_i$ is the goal or expected value, Y_i is the obtained value by the model and n is the total of samples tested. The values obtained in the experiments are shown in Table 2.

Table 2
MSE obtained in the experiments

Model	MSE
T1 RBFNN (Logistic RBF)	24.299
T1 RBFNN (Gaussian RBF)	0.9853
IT-2 RBFNN (Logistic RBF)	28.3849
IT-2 RBFNN (Gaussian RBF)	0.1974

The use of the logistic RBF as is shown in Fig. 9 demonstrates that this function produces a big error rate with an MSE value of 24.29 for T1 RBFNN and 28.3849 for IT2 RBFNN that increases the error with IT2 model (see fig. 10).

The classic Gaussian RBF provides good results for the IT2 model when it is compared to their counterpart of T1 (Fig. 12) and with the logistic RBFNN in both types reducing the error in a proportion of 24.66 times less for T1 and 143.609 times for IT2 models.

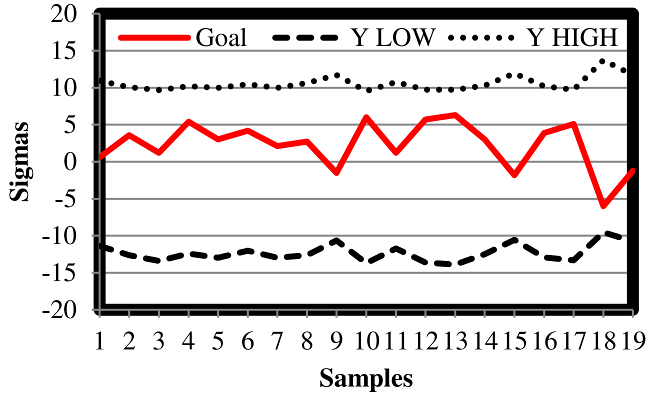

Fig. 8. IT2 RBFNN approximations with logistic RBF’s without type reduction.

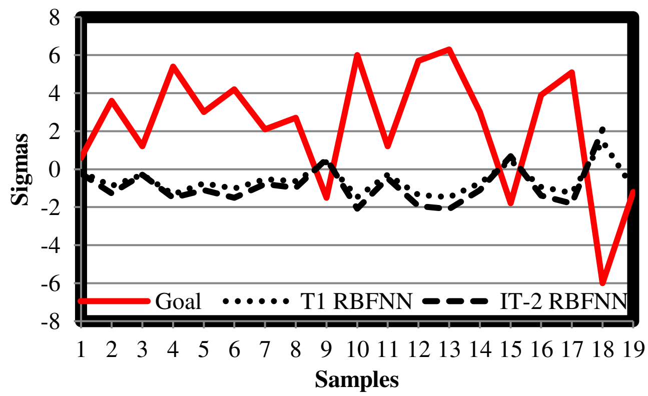

Fig. 9. RBFNN approximations with logistic RBF’s, IT2 RBFNN with type reduction.

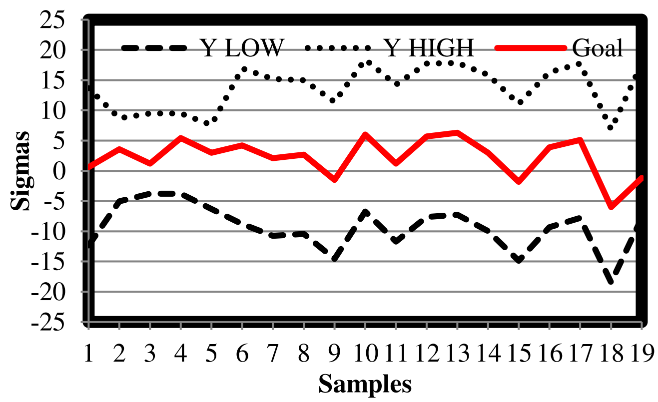

Fig. 10. IT2 RBFNN approximations with Gaussian RBF’s without type reduction.

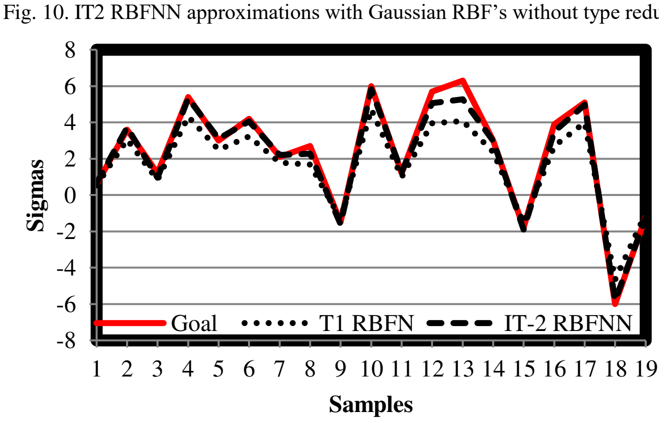

Fig. 11. RBFNN approximations with Gaussian RBF’s, IT2 RBFNN whit type reduction.

6. *Conclusion*

The use of CCD provides a better method to assemble the fuzzy rule base in a simplified and compact manner with the advantage of a compact base with a few rules that provide precise approximations as outputs.

A simplified and compact rule base reduces the computational times, expend in the calculations.
The use of the logistic RBF turned out not to be suitable for a network without training and this is part of future work.

The adapted model of Fig. 7 provides an accurate result without training.

The classic Gaussian RBF provides good results for the IT-2 model when it is compared to their counterpart of T1 with a five times better enhancement in the prediction.

The most important enhancement is the use of a RBFNN that does not need training and provides accurate results.

7. *Declarations*

Author contribution All authors contributed significantly to the work in accordance with the order provided.

Funding 'Not applicable'

Conflicts of interest/Competing interests The authors declare that they have no conflict of interest

Code availability 'Not applicable'

Ethics approval 'Not applicable'

Consent to participate 'Not applicable'

Consent for publication 'Not applicable'

References

[1] Mendel JM. (2001). Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Upper Saddle River, NJ: Prentice-Hall.

[2] Rubio-Solis A & Panoutsos, G. (2015) Interval type-2 radial basis function neural network: a modeling framework. IEEE Transactions on Fuzzy Systems, 23(2):457-473.

[3] Dorantes PNM, Nieto González J, and Méndez GM , (2014) Fault Detection Systems via a Novel Hybrid Methodology for Fuzzy Logic Systems Based on Individual Base Inference and Statistical Process Control, Latin Am. Trans., IEEE (Revista IEEE America Latina), 12(4):706–712.

[4] Macvicar-Whelan P J. (1978). Fuzzy Sets, the Concept of Height, and the Hedge very. *IEEE transactions on systems, man, and cybernetics,* 8(6)507-511.

[5] Dongale TD, Kulkarni TG, Kadam PA, Muldholkar RR. “Simplified Method for Compiling Rule Base Matrix”. International Journal of Soft Computing and Engineering (IJSCE). Vol. 2, No. 1, 2012, pp.

[6] Zarandi FMH, Alaeddini A, Turksen IB. (2008) A hybrid fuzzy adaptative sampling-Run rules for the Shewart control charts. *Information Sciences* 178:1152-1170.

[7] Serturk S & Erginel N. “Development of fuzzy X-R and X-S cntrol charts using α–cuts”. *Information Sciences* vol. 179. 2009, pp. 1542-1551.

[8] Gülbay M & Kahraman C. (2007) An alternative approach to fuzzy control charts: Direct fuzzy approach. *Information sciences,* 177:1463-1480.

[9] Dorantes PNM, Mexicano Santoyo A, Méndez GM. (2018) Modeling Type-1 Singleton Fuzzy Logic Systems using statistical parameters in foundry temperature control application. *Smart and Sustainable Manufacturing systems ASTM.* 2(1):180-203.

[10] Dorantes P N M and Méndez GM. (2015) Non-iterative Radial Basis Function Neural Networks to Quality Control via Image Processing, *IEEE LATIN AMERICA TRANSACTIONS*, 13(10): 3457-3451.

[11] Méndez GM, Dorantes PNM & Mexicano Santoyo A. (2019). Interval type-2 fuzzy logic systems optimized by central composite design to create a simplified fuzzy rule base in image processing for quality control application. *International Journal of advanced Manufacturing technology,* 102(9), 3757-3766.

[12] Rubio-Solis A, Melin P, Martinez-Hernandez U & Panoutsos, G. (2019). General Type-2 Radial Basis Function Neural Network: A Data-Driven Fuzzy Model. *IEEE Transactions on Fuzzy Systems*, *27(* 2):333-347.

[13] Tavoosi J, Suratgar A A & Menhaj MB., (2016) Nonlinear system identification based on a self-organizing type-2 fuzzy RBFN. *Engineering Applications of Artificial Intelligence*, 54:26-38.

[14] Ngo PD, & Shin YC. (2016), Modeling of unstructured uncertainties and robust controlling of nonlinear dynamic systems based on type-2 fuzzy basis function networks. *Engineering Applications of Artificial Intelligence*, 53:74-85.

[15] Baklouti N, Abraham A & Alimi AM.. (2018) A Beta basis function Interval Type-2 Fuzzy Neural Network for time series applications. Engineering Applications of Artificial Intelligence, 71:259–274.

[16] Jang RS, Sun C, Mizutani E., (1997) Neuro-Fuzzy and Soft Computing. A computational approach to learning and machine intelligence, Upper Saddle River, NJ: Prentice-Hall.

[17] Jang J S R and Sun C T (1993) Functional equivalence between radial basis function networks and fuzzy inference systems, *IEEE Trans. Neural Netw.*, 4(1):156–158.

[18] Andersen H C, Zadeh LA, and Westphal LC. (1998) Comments on functional equivalence between radial basis function networks and fuzzy inference systems,” *IEEE Trans. Neural Netw.*, 9(6):1529–1532.

[19] Hunt K J, Haas R,, and Murray-Smith R. (1996))Extending the functional equivalence between radial basis function networks and fuzzy inference systems, *IEEE Trans. Neural Netw.*, 7(3):776–781.

[20] Wang L.X.. (1996) A Course in Fuzzy Systems and control, Prentice Hall. 1st. Edition. .