

RESEARCH ARTICLE

ALTERNATIVE KARNIK-MENDEL ALGORITHM FOR INTERVAL TYPE-2 TYPE-REDUCTION

^{1,2}Gedler Gabriel, ¹Gedler Gustavo and ²Flaviani Federico¹Modeling and Software Development, MS2 Consulting Group, Caracas, Venezuela²Computing and Information Technology Department, Simón Bolívar University, Caracas, VenezuelaAccepted 18th May 2019; Published Online 30th June 2019

ABSTRACT

The Karnik-Mendel algorithm computes the centroids of an Interval Type-2 Fuzzy Set, corresponding to the left and right endpoints of the interval centroid. According to Mendel and WU (2009), the computation of those endpoints, represent a bottleneck for interval type-2 fuzzy logic systems, and they proposed a more efficient algorithm called the Enhanced Karnik-Mendel algorithm (2009). In this paper, the monotony property of the centroids is proven, and a new algorithm is suggested based on this property and the aforementioned Enhanced Karnik-Mendel algorithm, that improves computation times when the number of rules is roughly less than 1,000.

Key words: Centroid calculation, interval type-2 fuzzy system, Karnik-Mendel algorithm, Neuro Fuzzy Inference, Type Reduction.

INTRODUCTION

The fuzzy logic systems (FLSs), and in particular Interval Type-2 (IT2) decision systems, IT2 neural networks have been used in diverse fields such as mobile robots control [1], time-series forecasting [2], Image Processing Applications [3], predicting ATM demands [4], surrogate model generation [5], and numerous other applications. A Type-2 FLS allows for a better modelling of the uncertainty than a Type-1 FLS, because it has a Footprint of Uncertainty (FOU) that gives it more degrees of freedom [6]. An Interval Type-2 Fuzzy Set (IT2 FS) is a simplified version of the general Type-2 FS where its grade of membership is a constant interval rather than a function. It has been shown that IT2 FLSs can outperform their Type-1 FLSs counterparts in a variety of fields, such as information processing, fuzzy control and decision-making [6]. The computation of the centroid of an IT2 FS, developed originally by Karnik and Mendel [7], provides a measure of the uncertainty of that FS and is later used for type reduction. This is also one of the most important computations for that FS.

The Karnik-Mendel algorithm involves a procedure for computing the centroid of an IT2 FS, corresponding to the determination of the left and right endpoints of the interval centroid and denoted by L and R (the switch points found by the KM algorithm). This is an iterative algorithm and represents a bottleneck for the overall performance of the FS. Wu and Mendel optimized the original algorithm in the Enhanced Karnik-Mendel (EKM) algorithm [8]. IT2 FSs, and, in particular, fuzzy neural networks (also called neuro-fuzzy system), are composed of an Input Layer, a Fuzzification Layer, an Inference Layer (where the T-norm is applied), a Type-Reduction Layer and an Output Layer. In this work, proof that the centroid calculation function is a monotonic function is shown, and the EKM algorithm is presented, to then show a new alternative for calculating the centroid, based on the prior. Lastly, a comparison is done between the KM algorithm, the EKM algorithm and the alternative algorithm showing the improvement obtained by the new algorithm. This paper is organized as follows: first, the Karnik and Mendel solution for the discrete case and their algorithm are presented, and then a property of monotonicity of the centroid functions and a simplification of the criteria used in the algorithm are shown. After that, an alternative algorithm based on those properties is set up, and at the end, a comparison of both algorithms is carried out.

Karnik-mendel discrete version of centroid: Karnik and Mendel proved that when calculating the type-reduction of an IT2 FS, the centroids C_L and C_R can be computed from the lower and upper membership function $\check{A}(L)$ and $\check{A}(R)$ of a given Fuzzy Set \check{A} using the following equations:

$$C_L = \min_k C_L(k) = \min_{L \in \mathbb{N}} \text{centroid}(\check{A}(L))$$

$$C_R = \max_k C_R(k) = \max_{R \in \mathbb{N}} \text{centroid}(\check{A}(R))$$

Where

$$C_L(k) = \frac{\sum_{i=1}^k \underline{w}_i \bar{I}_i + \sum_{i=k+1}^N \underline{w}_i \underline{I}_i}{\sum_{i=1}^k \bar{I}_i + \sum_{i=k+1}^N \underline{I}_i}$$

$$C_R(k) = \frac{\sum_{i=1}^k \bar{w}_i \underline{I}_i + \sum_{i=k+1}^N \bar{w}_i \bar{I}_i}{\sum_{i=1}^k \underline{I}_i + \sum_{i=k+1}^N \bar{I}_i}$$
(1)

In the case of a fuzzy neural network, $\bar{I}_i, \underline{I}_i \in [0,1]$ represent the grade of membership of x_i to \check{A} , $x_i \in \mathbb{R}$, and \bar{w}_i respectively \underline{w}_i are their associated weights, whereas L is the switch point from \bar{I}_i to \underline{I}_i whereas R is the switch point from \underline{I}_i to \bar{I}_i (Figure 1).

N defines the number of points where the domain of \check{A} has been discretized. It is assumed that $(w_i)_{i=1}^N$, are increasingly ordered [9].

In addition, we denote $C_L(L) = C_L$ and $C_R(R) = C_R$ as the minimum and maximum centroids, respectively.

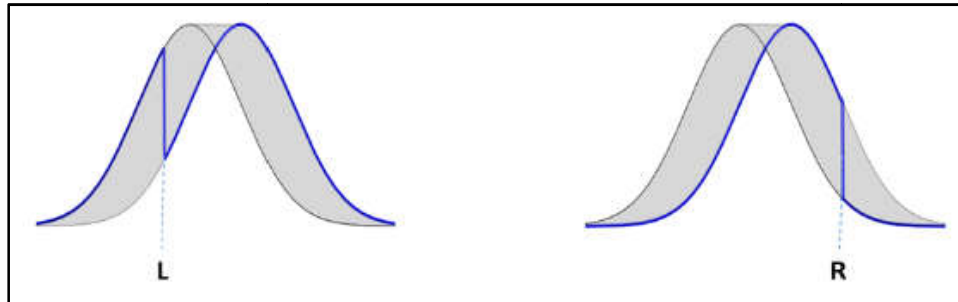


Figure 1. Definition of L and R in type-2 fuzzy set

Since L and R are unknown, an iterative algorithm that tries out the different values of C_L and C_R has to be done.

K-M Recursive Algorithm

Karnik and Mendel designed an algorithm that finds the switch points and centroids [7], called the KM algorithm. This algorithm was then improved by Wu and Mendel, called the EKM algorithm [8], and is shown here:

EKM Algorithm (Mendel & Wu)

1. Order the weights ω_i in ascending order, and associate each inferred rule I_i with its corresponding weight.
2. Set $k = \lceil n/2.4 \rceil$ (closest integer to $n/2.4$) and compute

$$a = \sum_{i=1}^k \bar{I}_i \underline{\omega}_i + \sum_{i=k+1}^N \underline{I}_i \underline{\omega}_i$$

$$b = \sum_{i=1}^k \bar{I}_i + \sum_{i=k+1}^N \underline{I}_i$$

$$y = \frac{a}{b}$$

1. Find k' such that $\underline{\omega}_{k'} \leq y \leq \underline{\omega}_{k'+1}$
2. If $k' = k$ stop the algorithm.
3. In other cases, compute

$$s = \text{sign}(k' - k)$$

$$a' = a + s \sum_{i=\min(k,k')+1}^{\max(k,k')} \underline{\omega}_i (\bar{I}_i - \underline{I}_i)$$

$$b' = b + s \sum_{i=\min(k,k')+1}^{\max(k,k')} (\bar{I}_i - \underline{I}_i)$$

$$y' = \frac{a'}{b'}$$

1. Set $y = y'$, $a = a'$, $b = b'$ $yk = k'$. And go to step 1.

At the end, k represents the value of L (or R , in the case of c_R), and y represents c_L (or c_R).

Monotonicity of the centroid functions

In this section an alternative algorithm for the calculation of c_L (and c_R), derived from (1), which reduces the computation times in relation to the KM and EKM algorithms generally used. In order to deduce this algorithm some propositions should be demonstrated before.

The following basic general relations can be shown,

If $b > 0$ and $d > 0$.

$$\frac{a}{b} \leq \frac{a+c}{b+d} \Leftrightarrow \frac{a}{b} \leq \frac{c}{d} \tag{2}$$

$$\frac{a+c}{b+d} \leq \frac{a}{b} \Leftrightarrow \frac{c}{d} \leq \frac{a}{b} \tag{3}$$

If $b > d > 0$

$$\frac{a}{b} \leq \frac{a-c}{b-d} \Leftrightarrow \frac{a}{b} \geq \frac{c}{d} \tag{4}$$

$$\frac{a-c}{b-d} \leq \frac{a}{b} \Leftrightarrow \frac{c}{d} \geq \frac{a}{b} \tag{5}$$

This new algorithm is based on the following proposition [5].

Proposition

If the weights associated to $c_L(k)$ and $c_R(k)$ are in ascending order, those functions are respectively concave and convex, and the minimum (maximum) is located on L (R).

$$\forall k < L, c_L(k-1) \geq c_L(k) \text{ and } \forall k \geq L, c_L(k) \leq c_L(k+1)$$

$$\forall k < R, c_R(k-1) \leq c_R(k) \text{ and } \forall k \geq R, c_R(k) \geq c_R(k+1)$$

Demonstration $c_L(k)$

Denoting the numerator and denominator of $c_L(k)$, by $\underline{a}(k)$, $\underline{b}(k)$ and defining $\Delta_k = \bar{I}_k - \underline{I}_k$, it can be shown that:

$$c_L(k) = \frac{\underline{a}(k)}{\underline{b}(k)} = \frac{\underline{a}(k-1) + \Delta_k w_k}{\underline{b}(k-1) + \Delta_k} = \frac{\underline{a}(k+1) - \Delta_{k+1} w_{k+1}}{\underline{b}(k+1) - \Delta_{k+1}}, \quad 1 < k < N \tag{6}$$

By definition, it is known that $\underline{b}(k)$, Δ_k and $\underline{b}(k) - \Delta_k$ are greater than zero.

Let's suppose without loss of generality that $(w_i)_{i=1}^N$ are increasingly ordered. Since $c_L(k)$ reaches its minimum on $k = L$, it is known that

$$c_L(L-1) \geq c_L(L) \leq c_L(L+1), \tag{7}$$

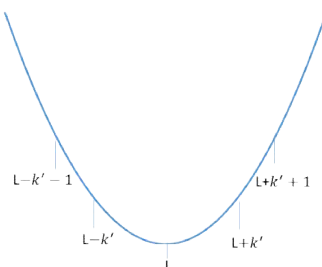


Figure 1. Behavior in both sides of the minimum

It is known from the definition of the centroid that the sum's domain is $[1, N]$, with L and R being within this domain. The demonstration will be done in two steps, one for $1 \leq k \leq L$, (left of L , Figure 2) and another for $L \leq k \leq N$ (right of L , Figure 2) thus ensuring that the successor is always defined within the domain described above. Using $k = L \mp k'$, $k' \geq 0$.

Case $k = L + k'$, with $k' \leq N - L$.

It will be shown the equivalent statement $c_L(L + k') \leq c_L(L + k' + 1)$, for $k' \leq N - L - 1$ using the mathematical induction technique over k' .

- When $k' = 0$, $c_L(L) \leq c_L(L + 1)$, because of (7), and the statement holds, shown the base case
- For the inductive step, let us suppose that for a given $k' \leq N - L - 2$, we have $c_L(L + k') \leq c_L(L + k' + 1)$ and it has to be demonstrated that $c_L(L + k' + 1) \leq c_L(L + k' + 2)$.

Using the induction hypothesis and the definition of $c_L(k)$, with $k = L + k'$

$$\frac{\underline{a}(k)}{\underline{b}(k)} = c_L(k) \leq c_L(k + 1) = \frac{\underline{a}(k + 1)}{\underline{b}(k + 1)}$$

By definition of $c_L(k)$, (6),

$$\frac{\underline{a}(k)}{\underline{b}(k)} = \frac{\underline{a}(k + 1) - \Delta_{k+1}w_{k+1}}{\underline{b}(k + 1) - \Delta_{k+1}} \leq \frac{\underline{a}(k + 1)}{\underline{b}(k + 1)}$$

Using (5) and the fact that w_i are increasingly ordered,

$$\frac{\underline{a}(k + 1)}{\underline{b}(k + 1)} \leq \frac{\Delta_{k+1}w_{k+1}}{\Delta_{k+1}} = w_{k+1} \leq w_{k+2} = \frac{\Delta_{k+2}w_{k+2}}{\Delta_{k+2}}$$

Then, using (2) and (6),

$$\frac{\underline{a}(k + 1)}{\underline{b}(k + 1)} \leq \frac{\underline{a}(k + 1) + \Delta_{k+2}w_{k+2}}{\underline{b}(k + 1) + \Delta_{k+2}} = \frac{\underline{a}(k + 2)}{\underline{b}(k + 2)}$$

Thus, $c_L(k + 1) \leq c_L(k + 2)$.

Case $k = L - k'$, with $k' \leq L - 1$.

In the same way using the mathematical induction technique over k' , it will be shown that:

$$c_L(L - k') \leq c_L(L - (k' + 1)) = c_L(L - k' - 1), \text{ for } k' < L - 1$$

- When $k' = 0$, $c_L(L) \leq c_L(L - 1)$ because of (7), and the statement holds, shown the base case
- Let us suppose that for a given $k' < L - 2$, we have $c_L(L - k') \leq c_L(L - (k' + 1))$, it has to be demonstrated that $c_L(L - (k' + 1)) \leq c_L(L - (k' + 2))$.

Using the induction hypothesis and (6), with $k = L - k'$

$$c_L(k) = \frac{\underline{a}(k)}{\underline{b}(k)} = \frac{\underline{a}(k - 1) + \Delta_k w_k}{\underline{b}(k - 1) + \Delta_k} \leq \frac{\underline{a}(k - 1)}{\underline{b}(k - 1)}$$

After (3) and the fact that w_j are increasingly ordered,

$$\frac{\Delta_{k-1}w_{k-1}}{\Delta_{k-1}} = w_{k-1} \leq w_k = \frac{\Delta_k w_k}{\Delta_k} \leq \frac{\underline{a}(k - 1)}{\underline{b}(k - 1)}$$

Using (4) and (6),

$$\frac{\underline{a}(k - 1)}{\underline{b}(k - 1)} \leq \frac{\underline{a}(k - 1) - \Delta_{k-1}w_{k-1}}{\underline{b}(k - 1) - \Delta_{k-1}} = \frac{\underline{a}(k - 2)}{\underline{b}(k - 2)}$$

Thus, $c_L(k - 1) \leq c_L(k - 2)$

$$\Leftrightarrow c_L(L - k' - 1) \leq c_L(L - k' - 2)$$

Demonstration for $c_R(k)$

In a similar way, denoting, $c_R(k) = \frac{\bar{a}(k)}{\bar{b}(k)}$ and Δ_k defined previously it can be written that:

$$c_R(k) = \frac{\bar{a}(k)}{\bar{b}(k)} = \frac{\bar{a}(k-1) - \Delta_k \bar{w}_k}{\bar{b}(k-1) - \Delta_k}$$

And (8)

$$c_R(k) = \frac{\bar{a}(k+1) + \Delta_{k+1} \bar{w}_{k+1}}{\bar{b}(k+1) + \Delta_{k+1}}$$

Using this formula, and a procedure similar to the one used above, the demonstration for $c_R(k)$ holds.

Simplification of the Decision Criterion

Based on the property shown above, the relationship between y and y' used on the Mendel & Wu's algorithm can be replaced by $D = s \cdot (b \cdot \omega_k - a) > 0$ when y and y' are given by successive values of k , without having to compute y (the division is not needed). To do that, it should be demonstrated that

$$s \cdot \frac{\underline{a}(k-1)}{\underline{b}(k-1)} \leq s \cdot \frac{\underline{a}(k)}{\underline{b}(k)} \Leftrightarrow s \cdot (\underline{b}(k) \cdot \omega_k - \underline{a}(k)) \geq 0$$

Where s is the sign function

$$s = \begin{cases} 1 & \text{if } k > L \text{ or } k < R \\ -1 & \text{if } k < L \text{ or } k > R \end{cases}$$

Demonstration

Case c_L

Case $L < k \leq N$

$$\frac{\underline{a}(k-1)}{\underline{b}(k-1)} \leq \frac{\underline{a}(k)}{\underline{b}(k)}$$

Using (6):

$$\frac{\underline{a}(k-1)}{\underline{b}(k-1)} = \frac{\underline{a}(k) - \Delta_k w_k}{\underline{b}(k) - \Delta_k} \leq \frac{\underline{a}(k)}{\underline{b}(k)}$$

Using (5):

$$\frac{\underline{a}(k)}{\underline{b}(k)} \leq \frac{\Delta_k w_k}{\Delta_k} = w_k \Leftrightarrow 0 \leq \underline{b}(k) w_k - \underline{a}(k)$$

Case $0 < k \leq L$

$$\frac{\underline{a}(k-1)}{\underline{b}(k-1)} \geq \frac{\underline{a}(k)}{\underline{b}(k)}$$

Using (6):

$$\frac{\underline{a}(k-1)}{\underline{b}(k-1)} = \frac{\underline{a}(k) - \Delta_k w_k}{\underline{b}(k) - \Delta_k} \geq \frac{\underline{a}(k)}{\underline{b}(k)}$$

Using (4)

$$\frac{\underline{a}(k)}{\underline{b}(k)} \geq \frac{\Delta_k w_k}{\Delta_k} = w_k \Leftrightarrow 0 \geq \underline{b}(k) w_k - \underline{a}(k)$$

Simplification for the case of c_R

In a similar way, using (8), it can be proven that for: $k < R$

$$\frac{\Delta_k w_k}{\Delta_k} = w_k \leq \frac{\bar{a}(k)}{\bar{b}(k)} \Leftrightarrow \bar{b}(k) \cdot w_k - \bar{a}(k) \leq 0$$

And for $k > R$

$$\frac{\bar{a}(k)}{\bar{b}(k)} \leq \frac{\Delta_k w_k}{\Delta_k} = w_k \Leftrightarrow 0 \leq \bar{b}(k) \cdot w_k - \bar{a}(k)$$

In conclusion, it can be set that:

If $k > L$ or $k < R$

$$\frac{\underline{a}(k-1)}{\underline{b}(k-1)} \leq \frac{\underline{a}(k)}{\underline{b}(k)} \Leftrightarrow \underline{b}(k)w_k - \underline{a}(k) \leq 0$$

If $k < L$ or $k > R$

$$\frac{\bar{a}(k)}{\bar{b}(k)} \leq \frac{\bar{a}(k-1)}{\bar{b}(k-1)} \Leftrightarrow 0 \leq \underline{b}(k)w_k - \underline{a}(k)$$

Using the sign function previously defined, the criterion can be written as

$$s * \frac{\bar{a}(k)}{\bar{b}(k)} \leq s * \frac{\bar{a}(k-1)}{\bar{b}(k-1)} \Leftrightarrow 0 \leq s * (\underline{b}(k)w_k - \underline{a}(k))$$

Using those properties, the new algorithm can be written.

Alternative Karnik-Mendel Algorithm

The general idea behind the algorithm is as follows: an initial approximation is found the same way that it is found when using the EKM algorithm, then a direction (sign) is arbitrarily chosen, and the difference between the initial approximation and the next step is computed. If the direction chosen is the correct one, then the algorithm continues until the minimum (maximum) is found. If the direction chosen is not the correct one, then the sign changes and the algorithm continues until the optimum is found

Alternative Karnik-Mendel Algorithm

Sort the weights ω_i in ascending order, and associate each inferred rule y_i with its corresponding weight.

1. Set $k = [n/2.4]$ (the integer closest to $n/2.4$) and $s = -1$
2. Set:

$$a = \sum_{i=1}^k \bar{I}_i \omega_i + \sum_{i=k+1}^n \underline{I}_i \omega_i$$

$$b = \sum_{i=1}^k \bar{I}_i + \sum_{i=k+1}^n \underline{I}_i$$

$$\Delta_k = (\underline{I}_k - \bar{I}_k)$$

$$D = s \cdot (b \cdot \omega_k - a)$$

3. If $D > 0$,

Set $s = 1$, $k = k + 1$ and

$$\Delta_k = -s \cdot (\underline{I}_k - \bar{I}_k)$$

$$D = s \cdot (b \cdot \omega_k - a)$$

4. While $D \leq 0$

{

$$a \leftarrow a + \Delta \cdot \omega_k$$

$$b \leftarrow b + \Delta$$

$$\begin{aligned}
 k &\leftarrow k + s \\
 \Delta_k &= -s \cdot (\underline{I}_k - \bar{I}_k) \\
 D &= s \cdot (b \cdot \omega_k - a)
 \end{aligned}$$

}

At the end of the algorithm, c_L will be given by a/b , and L will be k if the final value of s has a value of -1 , or $k - 1$ if s has a value of 1 .

Type-reduction algorithm comparison: In this section, a runtime comparison between the different type-reduction algorithms is done. These algorithms were run in a Dell Inspiron 5759, with Windows 10 64-bit, 12 GB of RAM, and an Intel Core i5-6200U 2.3GHz processor. In the table, a comparison of the average times (in nanoseconds) is shown. This was obtained from doing 1000 iterations for each rule size N .

Table 1. Average runtime comparison between the different type-reduction algorithms

		KM	EKM	Alternative
Number of rules (N)	20	1468.07	1085.79	572.42
	50	3054.93	2056.28	1053.18
	100	5616.22	3452.85	1823.83
	200	10492.68	6969.95	3122.6
	500	23895.21	13852.94	7878.92
	1000	42555.92	22869.43	17169.84

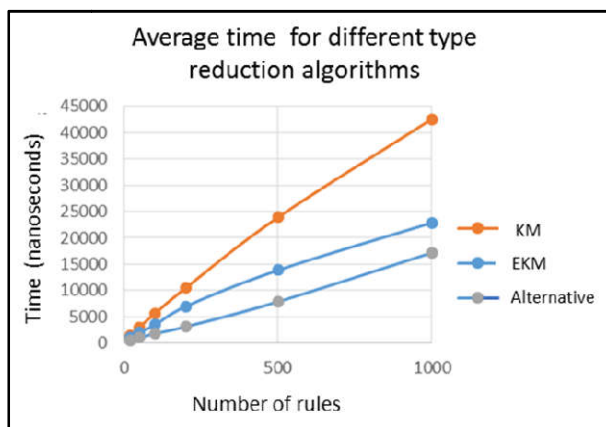


Figure 2. Runtime comparison between the type-reduction algorithms

The alternative algorithm reduces computational times considerably for small rule sets. As the number of rules increases, the alternative algorithm shows similar times to the EKM algorithm, and regression can show that eventually the EKM will be faster than the alternative algorithm (when the number of rules is larger than approximately 1,500). This is because the computational overhead caused by the division in the EKM algorithm eventually becomes lower than the computational overhead caused by doing an extremely large number of conditionals (ifs). Despite this, the biggest number of rules that Wu and Mendel propose is about 100, when calculating the centroid and variance of an IT2 fuzzy system [8].

Conclusion

The monotonic property of the centroids has been established. This property has been used to simplify the stop criterion in the EKM and develop an algorithm for finding the centroid of a type-2 FS that is faster than the previous algorithms presented in the literature. The times have been reduced to about half of the times of the EKM algorithm for small number of rules, and converge to the times of the EKM algorithm when the number of rules is close to 1,500.

REFERENCES

- [1] Baklouti, N. R. John, and A. Alimi; "Interval type-2 fuzzy logic control of mobile robots," Journal of Intelligent Learning Systems and Applications, 2012, 4, 291-302
- [2] Saima, H. J. Jaafar, S. Belhaouar, and T.A. Jillani, "ARIMA based interval type-2 fuzzy model for forecasting", International Journal of Computer Applications (0975 – 8887), Volume 28– No.3, August 2011
- [3] Castillo, O. M. A. Sanchez, C. I. Gonzalez; G. E. Martinez "Review of recent type-2 fuzzy image processing applications," Information August 2017, 97.
- [4] Darwish; S. M. "A methodology to improve cash demand forecasting for ATM network", International journal of Computer and Electrical Engineering, Vol. 5, No. 4, August 2013.

- [5] Gedler; G. “Aplicaciones de técnicas basadas en lógica difusa para la toma de decisiones” Informe de pasantía, Universidad Simón Bolívar, July 2017.
- [6] Liu X. “A Survey of continuous Karnik–Mendel algorithms and their generalizations,” *Advances in type-2 fuzzy sets and systems: theory and applications*, pp 19-31, edited by Alireza Sadeghian, J. M Mendel, H. Tahayori, June 2013
- [7] Karnik, N. N. and J. M. Mendel, “Centroid of a type-2 fuzzy set,” *Inf. Sci.*, vol. 132, pp. 195–220, 2001.
- [8] D. Wu, and J. M. Mendel “Enhanced Karnik–Mendel algorithms”, *IEEE Transaction on Fuzzy Systems*, VOL. 17, NO. 4, AUGUST 2009.
- [9] Mendel, J. M. and R. I. B. John, “Type-2 fuzzy set made simple”; *IEEE Transactions on Fuzzy Systems*, Vol. 10, No. 2, April 2000.
- [10] J. M. Mendel “Uncertain rule-based fuzzy logic systems: introduction and new directions.” Prentice-Hall Inc, Upper Saddle River, NJ (2001)
- [11] J. M. Mendel, “Interval type-2 fuzzy logic systems made simple”; *IEEE Transactions on Fuzzy Systems*, Vol. 14, No. 6, December 2006.
- [12] J. M. Mendel, “Type-2 fuzzy set and systems: an overview”; *IEEE Computational Intelligence Magazine*, February 2007
- [13] C.-H. Wang, C.-S. Cheng and T.-T. Lee; “Dynamical optimal training for interval type-2 fuzzy neural network (T2FNN)”, *IEEE Transaction on Systems, Man, and Cybernetic-part b: Cybernetics*, Vol. 14, No. 3, June 2004.
- [14] H.-J. Wu, Y.-L. Su, and S.-J. Lee, “A fast method for computing the centroid of a type-2 fuzzy set”; *IEEE Transaction on Systems, Man and Cybernetics- Part B: Cybernetics*, Vol. 42, No. 3, June 2012
