# Designing and Learning of Adjustable Quasi-Triangular Norms With Applications to Neuro-Fuzzy Systems

Leszek Rutkowski, *Fellow, IEEE,* and Krzysztof Cpałka

***Abstract*—In this paper, we introduce a new class of operators called quasi-triangular norms. They are denoted by $H$ and parameterized by a parameter $\nu: H(a_1, a_2, \ldots, a_n; \nu)$. From the construction of function $H$, it follows that it becomes a t-norm for $\nu = 0$ and a dual t-conorm for $\nu = 1$. For $\nu$ close to 0, function $H$ resembles a t-norm and for $\nu$ close to 1, it resembles a t-conorm. In the paper, we also propose adjustable quasi-implications and a new class of neuro-fuzzy systems. Most neuro-fuzzy systems proposed in the past decade employ "engineering implications" defined by a t-norm as the minimum or product. In our proposition, a quasi-implication $I(a, b; \nu)$ varies from an "engineering implication" $T\{a, b\}$ to corresponding S-implication as $\nu$ goes from 0 to 1. Consequently, the structure of neuro-fuzzy systems presented in this paper is determined in the process of learning. Learning procedures are derived and simulation examples are presented.**

***Index Terms*—Logical approach, Mamdani approach, neuro-fuzzy inference systems, triangular norms.**

## I. Introduction

TRIANGULAR norms (t-norms, and t-conorms called also s-norms in the engineering literature) have been used to model intersection and union of fuzzy sets, logical conjunction and disjunction and fuzzy preference (see e.g., [14] and [19]). For excellent surveys and overviews of various aggregation operators the reader is referred to [2], [5], [11], [14], [16], and [19]. Apart from traditional triangular norms several modifications and extensions have been proposed [2], [6], [11], [16]. Among those modifications we cite definitions of ordinal sums, uninorms, nullnorms and t-operators.

*Definition 1:* Suppose that $\{[a_i, b_i]\}_{i\in K}(a_i < b_i)$ is a countable family of nonoverlapping, closed subintervals of $[0,1]$, denoted by $\xi$. With each $[a_i, b_i] \in \xi$ associate a t-norm $T_i$. Let $T$ be a function defined on $[0,1]^2$ by

$$T\{x,y\} = \begin{cases} a_i + (b_i - a_i)T_i\left\{\frac{x-a_i}{b_i-a_i}, \frac{y-a_i}{b_i-a_i}\right\}, \\ \qquad \text{if } (x,y) \in [a_i, b_i]^2 \\ \min\{x,y\}, \text{ otherwise.} \end{cases} \tag{1}$$

Then, $T$ is called the ordinal sum of $\{([a_i, b_i], T_i)\}_{i\in K}$ and each $T_i$ is called a *summand*. It was shown [7] that $T$ is a t-norm.

*Definition 2:* A uninorm is a two-place function $U : [0,1] \times [0,1] \to [0,1]$ which is associative, commutative, nondecreasing in each place and there exists some element $e \in [0,1]$ called the neutral element such that $U(e,x) = x$ for all $x \in [0,1]$.

Obviously, for $e = 1$ function $U$ becomes a t-norm and for $e = 0$ a t-conorm. For any uninorm we have $U(0,1) \in \{0,1\}$.

*Definition 3:* A *nullnorm* is a two-place function $F : [0,1] \times [0,1] \to [0,1]$ which is associative, commutative, and nondecreasing in each place, and such that there exists some element $k \in [0,1]$ called the *absorbing element* satisfying $F(k,x) = k$ for all $x \in [0,1]$ and also $F(0,x) = x$ for all $x \le k$ and $F(1,x) = x$ for all $x \ge k$.

Note that Definition 3 is a generalization of t-norms and t-conorms. If $k = 0$, we obtain a t-norm; if $k = 1$, we get a t-conorm.

*Definition 4:* A *t-operator* is a two-place function $F : [0,1] \times [0,1] \to [0,1]$ which is associative, commutative, and nondecreasing in each place and such that

1) $F(0,0) = 0, F(1,1) = 1$;
2) sections $x \to F(x,0)$ and $x \to F(x,1)$ are continuous on $[0,1]$.

Definitions 1–4 are potentially useful to construct various new neuro-fuzzy systems, however in this paper we introduce another class of functions called quasi-triangular norms. They are denoted by $H$ and parameterized by parameter $\nu : H(a_1, a_2, \ldots, a_n; \nu)$. From the construction of function $H$ it follows that it becomes a t-norm for $\nu = 0$ and a dual t-conorm for $\nu = 1$. In this paper, we also propose adjustable quasi-implications. Most neuro-fuzzy systems proposed over the past decade [4], [8]–[10], [12], [13], [15], [17], [18], [21]–[24] employ "engineering implications" (this terminology is suggested in [17] and [18]) defined by a t-norm as the minimum or the product. As a matter of fact, the minimum or product "engineering implications" describe the conjunction operation between antecedents and consequences in Mamdani-type systems. However, it appears (see Section VI) that Mamdani-type systems are more suitable for approximation problems whereas for classification problems logical-type systems, based on $S, QL, R$, or other fuzzy implications [3], may be preferred. Therefore, in our proposition a quasi-implication $I(a, b; \nu)$ varies between an "engineering implication" $T\{a, b\}$ to corresponding S-implication as $\nu$ goes from 0 to 1. Moreover, it satisfies

$$I(a, b; \nu) = \begin{cases} I_{\text{eng}}(a,b) = T\{a,b\}, & \text{for } \nu = 0 \\ I_{\text{fuzzy}}(a,b) = S\{1-a, b\}, & \text{for } \nu = 1 \end{cases} \tag{2}$$

Manuscript received November 19, 2002; revised September 25, 2003 and March 19, 2004.

The authors are with the Department of Computer Engineering, Technical University of Częstochowa, 42-200 Częstochowa, Poland (e-mail: lrutko@kik.pcz.czest.pl).

Digital Object Identifier 10.1109/TFUZZ.2004.836069

assuming that $T$ and $S$ are dual triangular norms. It should be emphasized that parameter $\nu \in [0,1]$ can be learned from data. Consequently, we do not know in advance the type of the system. In this paper, we propose a new class of neuro-fuzzy systems. The fuzzy inference (Mamdani-type or logical-type) of such systems is determined in the process of learning. More precisely, after learning one of the following two possible structures is established depending on parameter $\nu$.

a) The neuro-fuzzy system with an "engineering implication" operator (t-norm) to describe the relation between the antecedents and the consequent, and with a t-conorm for the aggregation of rules.
b) The neuro-fuzzy system with an S-implication operator to describe the relation between the antecedents and the consequent, and with a t-norm for the aggregation of rules.

To the best of our knowledge, such a concept has not yet been proposed in literature by other authors. Neuro-fuzzy systems have been treated separately (Mamdani-type and logical-type) in a number of books and papers; see, e.g., [4], [19], and [22]. The type of a system has not yet been determined in the process of learning. The only relevance to our paper is the idea of Yager [29] who proposed the ordered weighted averaging (OWA) operator which becomes fuzzy max or min by assigning different values of the weighting vector. However, Yager's concept does not allow to learn a connectivity between the antecedent and the consequent.

The paper is organized as follows. In Section II, we introduce the concept of adjustable quasi-riangular norms. In Section III, we present adjustable quasi-implications. In Section IV, a new class of neuro-fuzzy systems is presented. In Section V, learning procedures are derived to learn parameter $\nu$ (type of the system) and parameters of membership functions. In Section VI, simulation examples are given.

## II. Adjustable Quasi-Triangular Norms

The results of this paper employ a concept of the compromise operator. A function

$$\tilde{N}_\nu : [0,1] \rightarrow [0,1] \tag{3}$$

given by

$$\begin{aligned}\tilde{N}_\nu(a) &= (1-\nu)N(a) + \nu N(N(a)) \\ &= (1-\nu)N(a) + \nu a\end{aligned} \tag{4}$$

is called a compromise operator where $\nu \in [0,1]$ and $N(a) = \tilde{N}_0(a) = 1 - a$.

Observe that $\tilde{N}_{1-\nu}(a) = \tilde{N}_\nu(1-a) = 1 - \tilde{N}_\nu(a)$ and

$$\tilde{N}_\nu(a) = \begin{cases} N(a), & \text{for } \nu = 0 \\ \frac{1}{2}, & \text{for } \nu = \frac{1}{2} \\ a, & \text{for } \nu = 1. \end{cases} \tag{5}$$

Obviously, function $\tilde{N}_\nu$ is a strong negation (see, e.g., [14]) for $\nu = 0$.

*Theorem 1:* Let $T$ and $S$ be dual triangular norms with respect to the classical fuzzy complement $N(a) = 1 - a$. The function $H$ mapping

$$H : [0,1]^n \rightarrow [0,1] \tag{6}$$

and given by

$$H(\mathbf{a};\nu) = \tilde{N}_\nu\left(S_{i=1}^n\{\tilde{N}_\nu(a_i)\}\right) = \tilde{N}_{1-\nu}\left(T_{i=1}^n\{\tilde{N}_{1-\nu}(a_i)\}\right) \tag{7}$$

varies from n the t-norm to the corresponding t-conorm as $\nu$ goes from 0 to 1.

*Proof:* From the assumption, it follows that

$$T\{\mathbf{a}\} = N(S\{N(a_1), N(a_2), \ldots, N(a_n)\}). \tag{8}$$

Formula (8) can be rewritten with notation of the compromise operator (4)

$$T\{\mathbf{a}\} = \tilde{N}_0(S\{\tilde{N}_0(a_1), \tilde{N}_0(a_2), \ldots, \tilde{N}_0(a_n)\}) \tag{9}$$

for $\nu = 0$. Apparently

$$S\{\mathbf{a}\} = \tilde{N}_1(S\{\tilde{N}_1(a_1), \tilde{N}_1(a_2), \ldots, \tilde{N}_1(a_n)\}) \tag{10}$$

for $\nu = 1$. □

The right-hand sides of (9) and (10) can be written as follows:

$$H(\mathbf{a};\nu) = \tilde{N}_\nu\left(S_{i=1}^n\{\tilde{N}_\nu(a_i)\}\right) \tag{11}$$

with $\nu = 0$ or $\nu = 1$. If parameter $\nu$ changes from 0 to 1, then the result is established.

The following definition comprises the main properties of an H-function.

*Definition 5:* An H-function $H : [0,1]^2 \rightarrow [0,1]$ is a function of two variables: $a_1, a_2$, and parameter $\nu$ that satisfies the following conditions:

H1) if $a_1 \leq a_3$ and $a_2 \leq a_4$,
then $H(a_1, a_2;\nu) \leq H(a_3, a_4;\nu)$,
for all $a_1, a_2, a_3, a_4, \nu \in [0,1]$
H2) $H(a_1, a_2;\nu) = H(a_2, a_1;\nu)$,
for all $a_1, a_2, \nu \in [0,1]$
H3) $H(a_1, H(a_2, a_3;\nu);\nu) = H(H(a_1, a_2;\nu), a_3;\nu)$,
for all $a_1, a_2, a_3 \in [0,1]$ and $\nu \in \{0,1\}$
H4) $H(1-\nu, a_1;\nu) = a_1$,
for all $a_1 \in [0,1]$ and $\nu \in \{0,1\}$.

Observe that the $H$-function, the t-norm, and the t-conorm are related to each other in the following way:

$$H(\mathbf{a};\nu) = \begin{cases} T\{\mathbf{a}\}, & \text{for } \nu = 0 \\ \frac{1}{2}, & \text{for } \nu = \frac{1}{2} \\ S\{\mathbf{a}\}, & \text{for } \nu = 1 \end{cases} \tag{12}$$

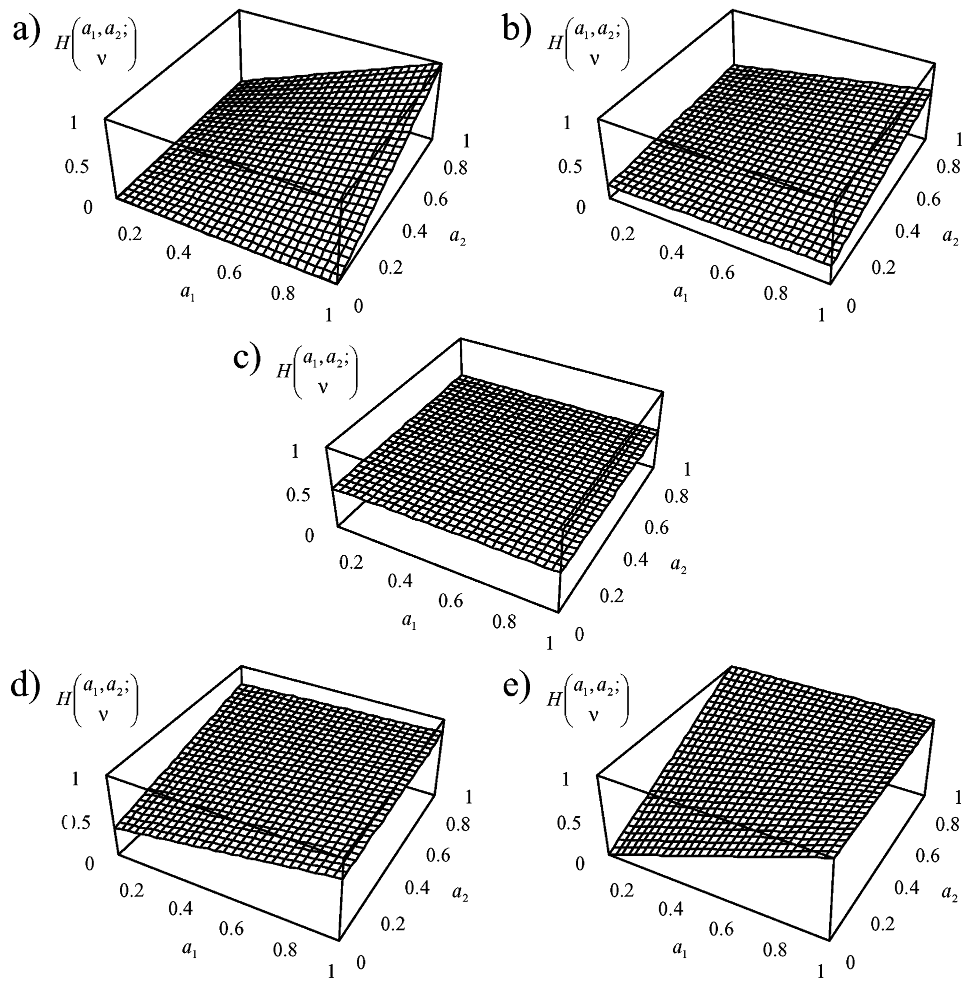


Fig. 1. Three-dimensional (3-D) plot of function (15) for (a) $\nu = 0.00$, (b) $\nu = 0.15$, (c) $\nu = 0.50$, (d) $\nu = 0.85$, and (e) $\nu = 1.00$.

It is easily seen, that for $0 < \nu < 0.5$ the H-function resembles a t-norm and for $0.5 < \nu < 1$ it resembles a t-conorm. The resemblance is more visible when parameter $\nu$ is close to 0 or to 1 (see Examples 1 and 2).

*Example 1 (Example of H-Function Generated by the Product t-Norm):* We will apply Theorem 1 to illustrate (for $n = 2$) how to switch between the product t-norm

$$\begin{aligned} T\{a_1, a_2\} &= H(a_1, a_2; 0) \\ &= a_1 a_2 \end{aligned} \tag{13}$$

and corresponding t-conorm

$$\begin{aligned} S\{a_1, a_2\} &= H(a_1, a_2; 1) \\ &= a_1 + a_2 - a_1 a_2. \end{aligned} \tag{14}$$

Following Theorem 1, the H-function generated by formula (13) or (14) is given by

$$\begin{aligned} H(a_1, a_2; \nu) &= \tilde{N}_{1-\nu}(\tilde{N}_{1-\nu}(a_1)\tilde{N}_{1-\nu}(a_2)) \\ &= \tilde{N}_{\nu}(1 - (1 - \tilde{N}_{\nu}(a_1))(1 - \tilde{N}_{\nu}(a_2))) \end{aligned} \tag{15}$$

and varies from (13) to (14) as $\nu$ goes from 0 to 1. In Fig. 1, we depict function (15) for: a) $\nu = 0.00$, b) $\nu = 0.15$, c) $\nu = 0.50$, d) $\nu = 0.85$, and e) $\nu = 1.00$.

*Example 2 (Example of H-Function Generated by the Łukasiewicz t-Norm):* For the Łukasiewicz t-norm

$$\begin{aligned} T\{a_1, a_2\} &= H(a_1, a_2; 0) \\ &= \max\{a_1 + a_2 - 1, 0\} \end{aligned} \tag{16}$$

and corresponding t-conorm

$$\begin{aligned} S\{a_1, a_2\} &= H(a_1, a_2; 1) \\ &= \min\{a_1 + a_2, 1\} \end{aligned} \tag{17}$$

the H-function generated by (16) or (17) takes the form

$$\begin{aligned} H(a_1, a_2; \nu) &= \tilde{N}_{1-\nu}(\max\{\tilde{N}_{1-\nu}(a_1) + \tilde{N}_{1-\nu}(a_2) - 1, 0\}) \\ &= \tilde{N}_{\nu}(\min\{\tilde{N}_{\nu}(a_1) + \tilde{N}_{\nu}(a_2), 1\}) \end{aligned} \tag{18}$$

and varies from (16) to (17) as $\nu$ goes from 0 to 1. In Fig. 2, we depict function (18) for: a) $\nu = 0.00$, b) $\nu = 0.15$, c) $\nu = 0.50$, d) $\nu = 0.85$, e) $\nu = 1.00$.

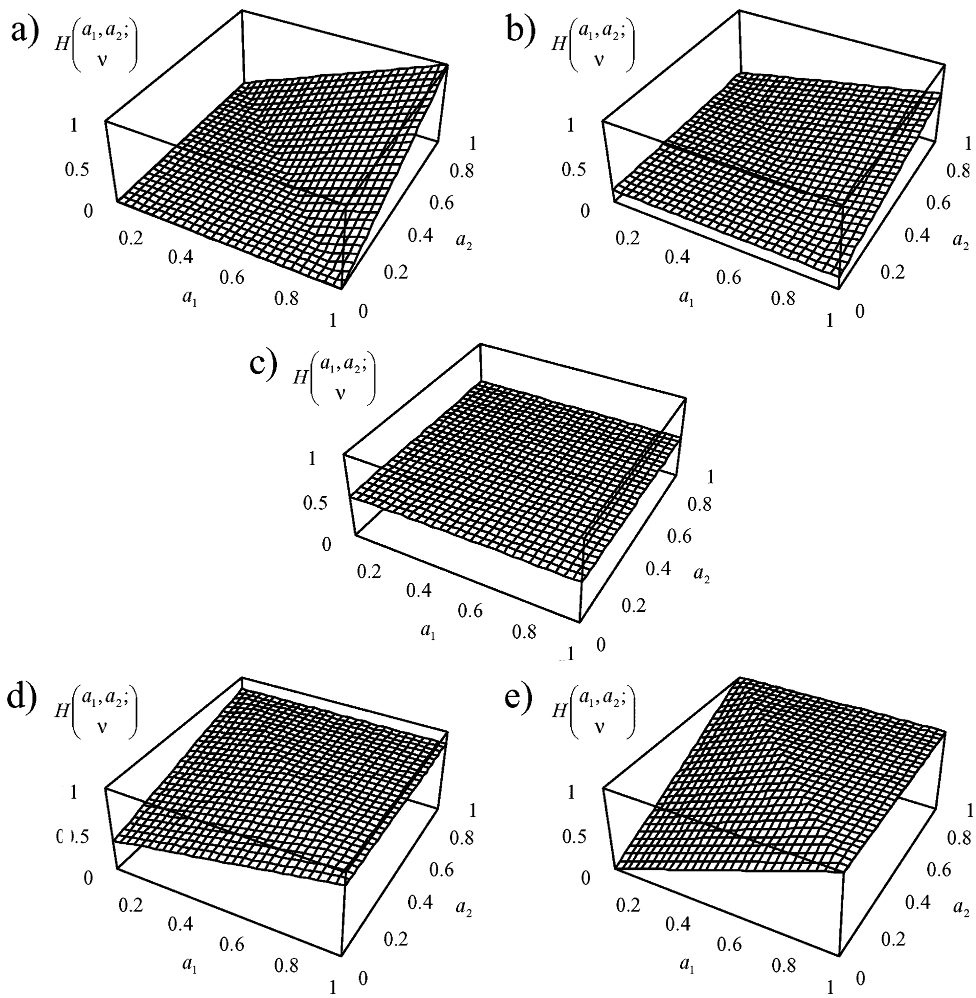


Fig. 2. 3-D plot of function (18) for (a) $\nu = 0.00$, (b) $\nu = 0.15$, (c) $\nu = 0.50$, (d) $\nu = 0.85$, and (e) $\nu = 1.00$.

## III. Adjustable Quasi-Implications

The next theorem shows how to switch between an "engineering implication" (defined by a t-norm) and an S-implication.

*Theorem 2:* Let $T$ and $S$ be dual triangular norms. Then

$$I(a, b; \nu) = H(\tilde{N}_{1-\nu}(a), b; \nu) \tag{19}$$

varies from "engineering implication"

$$\begin{aligned} I_{\text{eng}}(a, b) &= I(a, b; 0) \\ &= T\{a, b\} \end{aligned} \tag{20}$$

to corresponding S-implication

$$\begin{aligned} I_{\text{fuzzy}}(a, b) &= I(a, b; 1) \\ &= S\{1 - a, b\}. \end{aligned} \tag{21}$$

*Proof:* Theorem 2 is a straightforward consequence of Theorem 1. □

*Example 3 (Example of H-Implication Generated by the Product t-Norm):* Let

$$\begin{aligned} I_{\text{eng}}(a, b) &= H(a, b; 0) \\ &= T\{a, b\} \\ &= ab \end{aligned} \tag{22}$$

and

$$\begin{aligned} I_{\text{fuzzy}}(a, b) &= H(\tilde{N}_0(a), b; 1) \\ &= S\{N(a), b\} \\ &= 1 - a + ab. \end{aligned} \tag{23}$$

Then, (19) varies from the "engineering implication" represented by the product t-norm (22) to corresponding Reichenbach fuzzy implication (23) as $\nu$ goes from 0 to 1. This fact is illustrated in Fig. 3.

*Example 4 (Example of H-Implication Generated by the Łukasiewicz t-Norm):* Let

$$\begin{aligned} I_{\text{eng}}(a, b) &= H(a, b; 0) \\ &= T\{a, b\} \\ &= \max\{a + b - 1, 0\} \end{aligned} \tag{24}$$

and

$$\begin{aligned} I_{\text{fuzzy}}(a, b) &= H(\tilde{N}_0(a), b; 1) \\ &= S\{N(a), b\} \\ &= \min\{1 - a + b, 1\}. \end{aligned} \tag{25}$$

Then, (19) varies between the "engineering implication" represented by the Łukasiewicz t-norm (24) and Łukasiewicz fuzzy

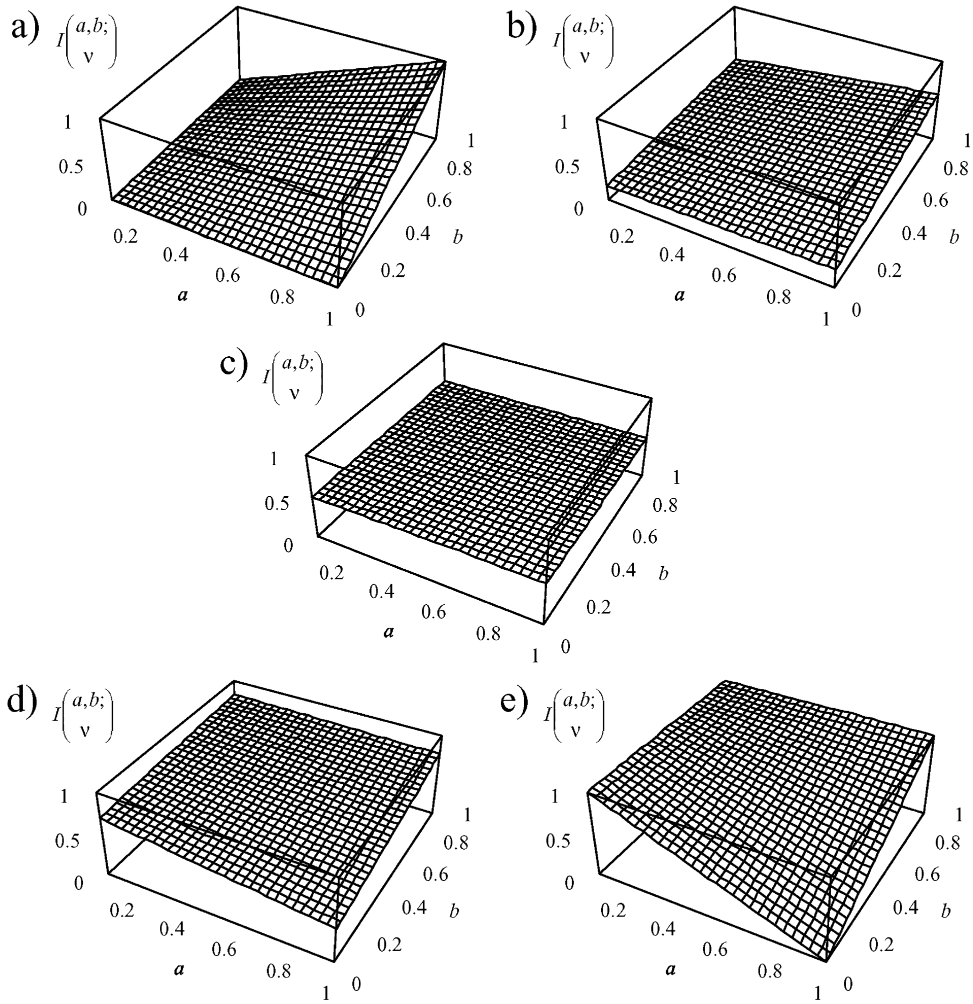


Fig. 3. 3-D plot of function (19) for the product generator for (a) $\nu = 0.00$, (b) $\nu = 0.15$, (c) $\nu = 0.50$, (d) $\nu = 0.85$, and (e) $\nu = 1.00$.

implication (25) as $\nu$ goes from 0 to 1. This fact is illustrated in Fig. 4.

## IV. Applications to Neuro-Fuzzy Systems

In this paper, we consider multiple-input–single-output fuzzy NFIS mapping $\mathbf{X} \to \mathbf{Y}$, where $\mathbf{X} \subset \mathbf{R}^n$ and $\mathbf{Y} \subset \mathbf{R}$.

The fuzzifier performs a mapping from the observed crisp input space $\mathbf{X} \subset \mathbf{R}^n$ to the fuzzy sets defined in $\mathbf{X}$. The most commonly used fuzzifier is the singleton fuzzifier which maps $\bar{\mathbf{x}} = [\bar{x}_1, \ldots, \bar{x}_n] \in \mathbf{X}$ into a fuzzy set $A' \subseteq \mathbf{X}$ characterized by the membership function

$$\mu_{A'}(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} = \bar{\mathbf{x}} \\ 0, & \text{if } \mathbf{x} \neq \bar{\mathbf{x}}. \end{cases} \tag{26}$$

The fuzzy rule base consists of a collection of $N$ fuzzy IF–THEN rules in the form

$$R^{(k)} : \begin{cases} \text{IF} & x_1 \text{ is } A_1^k \text{ AND} \\ & x_2 \text{ is } A_2^k \text{ AND} \ldots \\ & x_n \text{ is } A_n^k, \\ \text{THEN} & y \text{ is } B^k \end{cases} \tag{27}$$

or

$$R^{(k)} : \text{IF } \mathbf{x} \text{ is } A^k, \text{ THEN } y \text{ is } B^k \tag{28}$$

where $\mathbf{x} = [x_1, \ldots, x_n] \in \mathbf{X}, y \in \mathbf{Y}, A_1^k, A_2^k, \ldots, A_n^k$ are fuzzy sets characterized by membership functions $\mu_{A_i^k}(x_i)$, whereas $B^k$ are fuzzy sets characterized by membership functions $\mu_{B^k}(y)$, respectively, $k = 1, \ldots, N$.

The fuzzy inference determines a mapping from the fuzzy sets in the input space $\mathbf{X}$ to the fuzzy sets in the output space $\mathbf{Y}$. Each of $N$ rules (27) determines a fuzzy set $\bar{B}^k \subseteq \mathbf{Y}$ given by the compositional rule of inference

$$\bar{B}^k = A' \circ (A^k \to B^k) \tag{29}$$

where $A^k = A_1^k \times A_2^k \times \cdots \times A_n^k$ and

$$\begin{aligned} \tau_k(\bar{\mathbf{x}}) &= \mu_{A^k}(\bar{\mathbf{x}}) \\ &= T_{i=1}^n \mu_{A_i^k}(\bar{\mathbf{x}}) \end{aligned} \tag{30}$$

is a firing strength of the $k$th rule.

Fuzzy sets $\bar{B}^k$, according to (29), are characterized by membership functions expressed by the *sup-star* composition

$$\mu_{\bar{B}^k}(y) = \sup_{\mathbf{x} \in \mathbf{X}} \left\{ \mu_{A'}(\mathbf{x}) * \mu_{A_1^k \times \cdots \times A_n^k \to B^k}(\mathbf{x}, y) \right\} \tag{31}$$

where * can be any operator in the class of t-norms. It is easily seen that for a crisp input $\bar{\mathbf{x}} \in \mathbf{X}$, i.e., a singleton fuzzifier (26), formula (31) becomes

$$\begin{aligned} \mu_{\bar{B}^k}(y) &= \mu_{A_1^k \times \cdots \times A_n^k \to B^k}(\bar{\mathbf{x}}, y) \\ &= \mu_{A^k \to B^k}(\bar{\mathbf{x}}, y) \\ &= I(\mu_{A^k}(\bar{\mathbf{x}}), \mu_{B^k}(y)) \end{aligned} \tag{32}$$

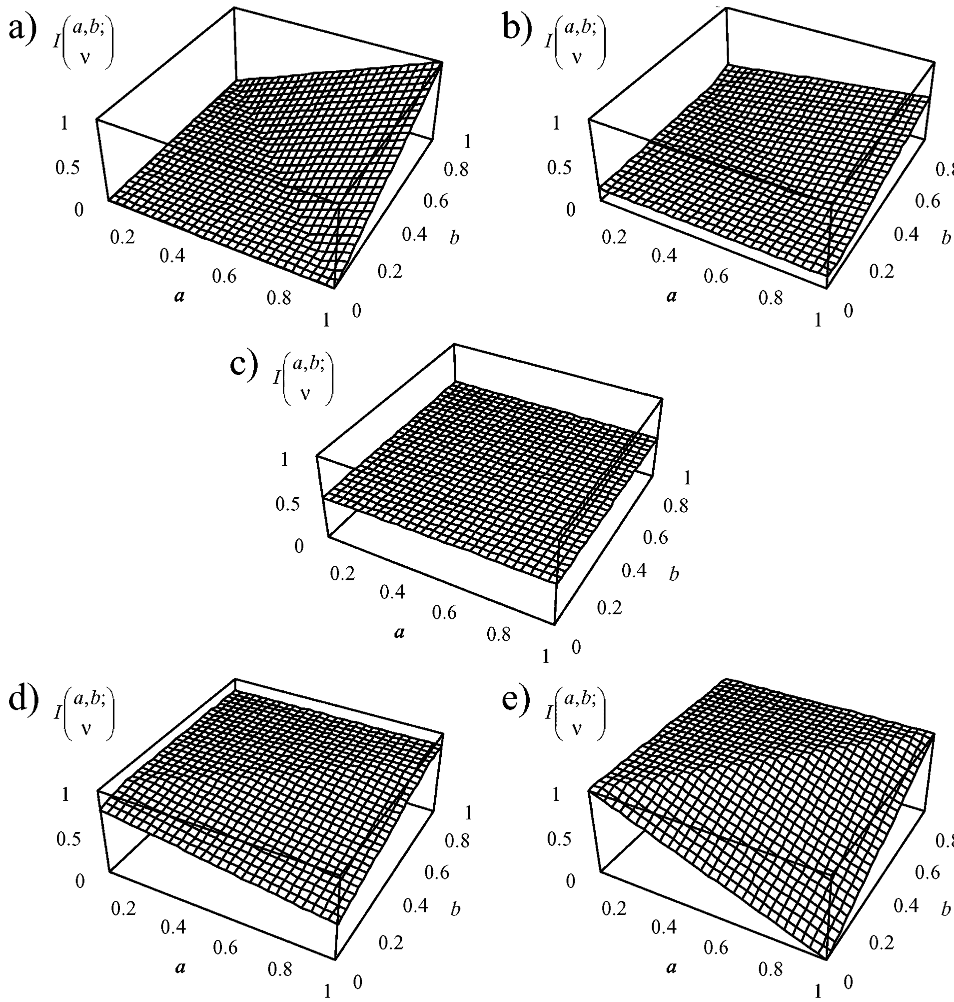


Fig. 4. 3-D plot of function (19) for the Łukasiewicz generator for (a) $\nu = 0.00$, (b) $\nu = 0.15$, (c) $\nu = 0.50$, (d)$\nu = 0.85$, and (e) $\nu = 1.00$.

where $I(\cdot)$ is an "engineering implication" or fuzzy implication. We note that fuzzy implications should satisfy the conditions of Fodor's lemma [5] and the most known in literature (see e.g., [3] and [4]) are S-implications, R-implications, and QL-implications, as we have already mentioned in Section I.

The aggregation operator, applied in order to obtain the fuzzy set $B'$ based on fuzzy sets $\bar{B}^k$, is the t-norm or t-conorm operator, depending on the type of fuzzy implication.

The defuzzifier performs a mapping from a fuzzy set $B'$ to a crisp point $\bar{y}$ in $\mathbf{Y} \subset \mathbf{R}$. The center of area (COA) method is defined by the following formula:

$$\bar{y} = \frac{\int_{\mathbf{Y}} y\mu_{B'}(y)\,dy}{\int_{\mathbf{Y}} \mu_{B'}(y)\,dy} \tag{33}$$

or by

$$\bar{y} = \frac{\sum_{r=1}^{N} \bar{y}^r \cdot \mu_{B'}(\bar{y}^r)}{\sum_{r=1}^{N} \mu_{B'}(\bar{y}^r)} \tag{34}$$

in the discrete form, where $\bar{y}^r$ denotes centers of the membership functions $\mu_{B^r}(y)$, i.e., for $r = 1, \ldots, N$

$$\mu_{B^r}(\bar{y}^r) = \max_{y \in \mathbf{Y}}\{\mu_{B^r}(y)\}. \tag{35}$$

*Theorem 3:* Let $T$ and $S$ be dual triangular norms. Then, the neuro-fuzzy system given by

$$\tau_k(\bar{\mathbf{x}}) = H\left(\mu_{A_1^k}(\bar{x}_1), \ldots, \mu_{A_n^k}(\bar{x}_n); 0\right) \tag{36}$$

$$I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r) = H(\tilde{N}_{1-\nu}(\tau_k(\bar{\mathbf{x}})), \mu_{B^k}(\bar{y}^r); \nu) \tag{37}$$

$$\mathrm{agr}_r(\bar{\mathbf{x}}, \bar{y}^r) = H(I_{1,r}(\bar{\mathbf{x}}, \bar{y}^r), \ldots, I_{N,r}(\bar{\mathbf{x}}, \bar{y}^r); 1-\nu) \tag{38}$$

$$\bar{y} = \frac{\sum_{r=1}^{N} \bar{y}^r \cdot \mathrm{agr}_r(\bar{\mathbf{x}}, \bar{y}^r)}{\sum_{r=1}^{N} \mathrm{agr}_r(\bar{\mathbf{x}}, \bar{y}^r)} \tag{39}$$

varies between Mamdani inference ($\nu = 0$) and logical inference ($\nu = 1$) as $\nu$ goes from 0 to 1.

*Proof:* For $\nu = 0$, (39) takes the form

$$\bar{y} = \frac{\sum_{r=1}^{N} \bar{y}^r \cdot S_{k=1}^{N}\{T\{\tau_k(\bar{\mathbf{x}}), \mu_{B^k}(\bar{y}^r)\}\}}{\sum_{r=1}^{N} S_{k=1}^{N}\{T\{\tau_k(\bar{\mathbf{x}}), \mu_{B^k}(\bar{y}^r)\}\}} \tag{40}$$

It is easily seen that inference in the $k$th rule is realized by a t-norm, whereas aggregation of individual rules by a t-conorm. For $\nu = 1$ procedure (39) becomes

$$\bar{y} = \frac{\sum_{r=1}^{N} \bar{y}^r \cdot T_{k=1}^{N}\{S\{N(\tau_k(\bar{\mathbf{x}}))), \mu_{B^k}(\bar{y}^r)\}\}}{\sum_{r=1}^{N} T_{k=1}^{N}\{S\{N(\tau_k\bar{\mathbf{x}}), \mu_{B^k}(\bar{y}^r)\}\}}. \tag{41}$$

TABLE I
IMPLICATION AND AGGREGATION OPERATORS AS $\nu$ VARIES FROM 0 TO 1

| Parameter $\nu$ | Implication | Aggregation |
|---|---|---|
| $\nu=0$ | $T\{a,b\}$ | t-conorm |
| $\nu=1$ | $S\{1-a,b\}$ | t-norm |
| $0<\nu<1$ | $H(\tilde{N}_{1-\nu}(a),b;\nu)$ | $H(a,b;1-\nu)$ |
| $\nu=0.5$ | $H(a,b;0.5)=0.5$ | $H(a,b;0.5)=0.5$ |

Therefore the inference in the $k$th rule is realized by an S-implication, whereas the aggregation of individual rules by a t-norm. Implication and aggregation operators in procedure (39) are shown in Table I for $\nu \in [0,1]$. □

## V. LEARNING PROCEDURES

Based on the learning sequence we wish to determine both parameters of input–output membership functions and parameter $\nu$ in the fuzzy system described by (39). Starting from an arbitrary initial value $\nu(0) \in [0,1]$ the system will adopt to the data and finally establish its structure as Mamdani-type (40) or logical-type (41). The scheme of (39) is depicted in Fig. 5. The errors propagated through the net are indicated at the bottom of Fig. 5.

*Remark 1:* We will explain the notation used in this section on a simple example of a single neuron given by

$$y=f(s), s=\sum_{i=1}^{n} x_i w_i$$

where $f$ is a sigmoidal function, $x_i$ and $w_i, i=1,\ldots,n$, are inputs and weights, respectively. Let $d$ be the desired output signal. Then, we write

$$\varepsilon^f=\varepsilon=y-d$$

and

$$\begin{aligned}\varepsilon^s &= \varepsilon^f\{s\}\\ &= \varepsilon^f \frac{\partial f(s)}{\partial s}\\ &= (y-d)f'(s)\end{aligned}$$

i.e., $\varepsilon^f\{s\}$ is the error transferred from the output block $f$ to the summation block $s$. We use the analogous notation depicted in Fig. 5.

### A. General Procedures

Parameter $\nu$ and parameters of input–output membership functions are updated by the recursive procedures

$$\nu(t+1)=\nu(t)-\eta\Delta\nu(t) \tag{42}$$
$$\sigma_{i,k}^A(t+1)=\sigma_{i,k}^A(t)-\eta\Delta\sigma_{i,k}^A(t) \tag{43}$$
$$\bar{x}_i^k(t+1)=\bar{x}_i^k(t)-\eta\Delta\bar{x}_i^k(t) \tag{44}$$
$$\sigma_k^B(t+1)=\sigma_k^B(t)-\eta\Delta\sigma_k^B(t) \tag{45}$$
$$\bar{y}^r(t+1)=\bar{y}^r(t)-\eta\Delta\bar{y}^r(t) \tag{46}$$

where corresponding corrections $\Delta$ are given by

$$\Delta\nu=\sum_{k=1}^{N}\sum_{r=1}^{N}\varepsilon_{k,r}^I\{\nu\}+\sum_{r=1}^{N}\varepsilon_r^{\mathrm{agr}}\{\nu\} \tag{47}$$
$$\Delta\sigma_{i,k}^A=\varepsilon_k^\tau\left\{\sigma_{i,k}^A\right\} \tag{48}$$

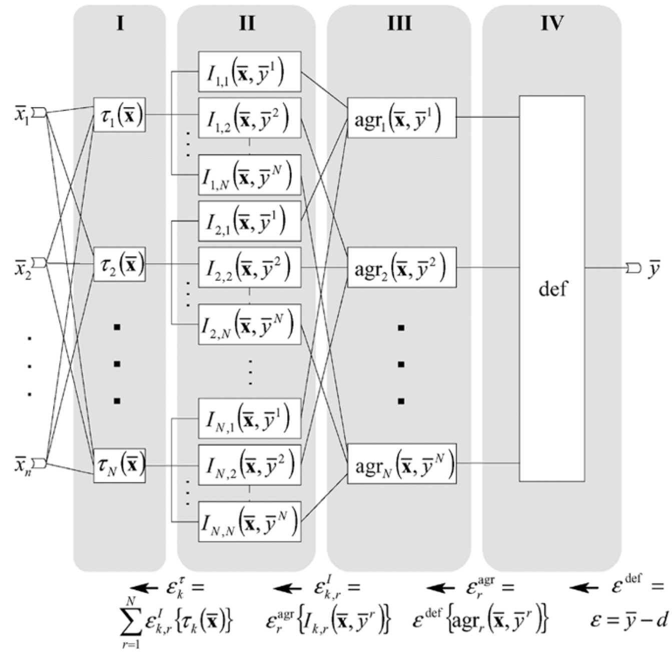


Fig. 5. Scheme of (39).

$$\Delta\bar{x}_i^k=\varepsilon_k^\tau\left\{\bar{x}_i^k\right\} \tag{49}$$
$$\Delta\sigma_k^B=\sum_{r=1}^{N}\varepsilon_{k,r}^I\left\{\sigma_k^B\right\} \tag{50}$$
$$\begin{aligned}\Delta\bar{y}^r &= \sum_{k=1}^{N}\varepsilon_{k,r}^I\{\bar{y}^r\}+\sum_{k=1}^{N}\varepsilon_{r,k}^I\{\bar{y}^r\}+\varepsilon^{\mathrm{def}}\{\bar{y}^r\}\\ &= \sum_{k=1}^{N}\left(\varepsilon_{k,r}^I\{\bar{y}^r\}+\varepsilon_{r,k}^I\{\bar{y}^r\}\right)+\varepsilon^{\mathrm{def}}\{\bar{y}^r\}.\end{aligned} \tag{51}$$

The errors propagated through the net are indicated at the bottom of Fig. 5 and satisfy the following relations:

$$\varepsilon_k^\tau=\sum_{r=1}^{N}\varepsilon_{k,r}^I\{\tau_k(\bar{\mathbf{x}})\} \tag{52}$$
$$\varepsilon_{k,r}^I=\varepsilon_r^{\mathrm{agr}}\{I_{k,r}(\bar{\mathbf{x}},\bar{y}^r)\} \tag{53}$$
$$\varepsilon_r^{\mathrm{agr}}=\varepsilon^{\mathrm{def}}\{\mathrm{agr}_r(\bar{\mathbf{x}},\bar{y}^r)\} \tag{54}$$
$$\varepsilon^{\mathrm{def}}=\varepsilon=\bar{y}-d. \tag{55}$$

### B. Block of Rules' Activation

The errors propagated through the block of rules' activation (see Fig. 6) are given by

$$\varepsilon_k^\tau\left\{\sigma_{i,k}^A\right\}=\varepsilon_k^\tau\frac{\partial\tau_k(\bar{\mathbf{x}})}{\partial\mu_{A_i^k}(\bar{x}_i)}\frac{\partial\mu_{A_i^k}(\bar{x}_i)}{\partial\sigma_{i,k}^A} \tag{56}$$
$$\varepsilon_k^\tau\left\{\bar{x}_i^k\right\}=\varepsilon_k^\tau\frac{\partial\tau_k(\bar{\mathbf{x}})}{\partial\mu_{A_i^k}(\bar{x}_i)}\frac{\partial\mu_{A_i^k}(\bar{x}_i)}{\partial\bar{x}_i^k} \tag{57}$$

where

$$\frac{\partial\tau_k(\bar{\mathbf{x}})}{\partial\mu_{A_i^k}(\bar{x}_i)}=\frac{\partial}{\partial\mu_{A_i^k}(\bar{x}_i)}H\left(\mu_{A_1^k}(\bar{x}_1),\ldots,\mu_{A_n^k}(\bar{x}_n);0\right). \tag{58}$$

The exact form of (58) and other derivatives depends on the used H-function and input–output membership functions. Some examples are given in the Appendix.

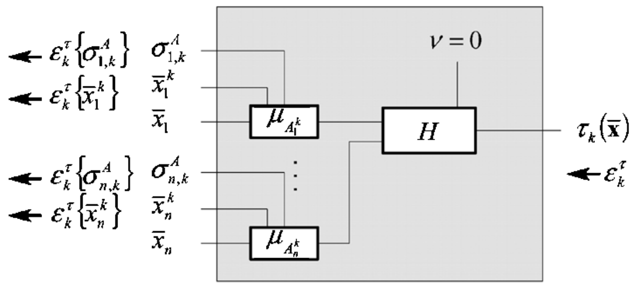


Fig. 6. Block of rules' activation.

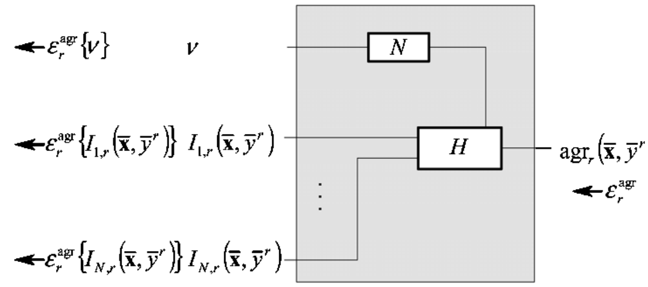


Fig. 8. Block of aggregation.

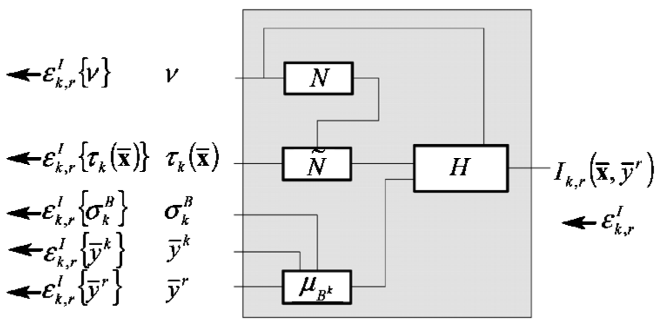


Fig. 7. Block of implications.

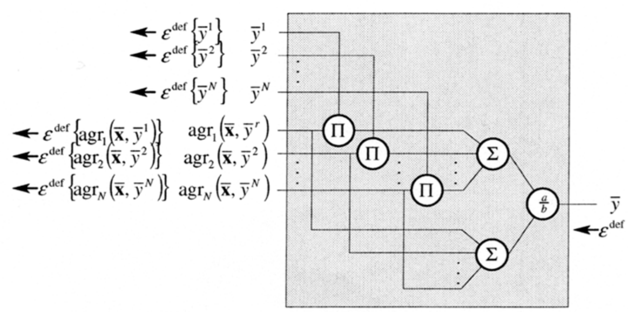

Fig. 9. Block of defuzzification.

## C. Block of Implications

The errors propagated through the block of implications (see Fig. 7) are determined as follows:

$$\varepsilon_{k,r}^{I}\{\nu\} = \varepsilon_{k,r}^{I}\left(\frac{\partial I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r)}{\partial \nu} + \frac{\partial I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r)}{\partial \tilde{N}_{1-\nu}(\tau_k(\bar{\mathbf{x}}))} \frac{\partial \tilde{N}_{1-\nu}(\tau_k(\bar{\mathbf{x}}))}{\partial (1-\nu)} \frac{\partial N(\nu)}{\partial \nu}\right) \tag{59}$$

$$\varepsilon_{k,r}^{I}\left\{\sigma_k^B\right\} = \varepsilon_{k,r}^{I} \frac{\partial I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r)}{\partial \mu_{B^k}(\bar{y}^r)} \frac{\partial \mu_{B^k}(\bar{y}^r)}{\partial \sigma_k^B} \tag{60}$$

$$\varepsilon_{k,r}^{I}\{\bar{y}^r\} = \varepsilon_{k,r}^{I} \frac{\partial I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r)}{\partial \mu_{B^k}(\bar{y}^r)} \frac{\partial \mu_{B^k}(\bar{y}^r)}{\partial \bar{y}^r} \tag{61}$$

$$\varepsilon_{k,r}^{I}\{\tau_k(\bar{\mathbf{x}})\} = \varepsilon_{k,r}^{I} \frac{\partial I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r)}{\partial \tilde{N}_{1-\nu}(\tau_k(\bar{\mathbf{x}}))} \frac{\partial \tilde{N}_{1-\nu}(\tau_k(\bar{\mathbf{x}}))}{\partial \tau_k(\bar{\mathbf{x}})} \tag{62}$$

where

$$\frac{\partial I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r)}{\partial \nu} = \frac{\partial}{\partial \nu} H(\tilde{N}_{1-\nu}(\tau_k(\bar{\mathbf{x}})), \mu_{B^k}(\bar{y}^r); \nu) \tag{63}$$

$$\frac{\partial I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r)}{\partial \tilde{N}_{1-\nu}(\tau_k(\bar{\mathbf{x}}))} = \frac{\partial}{\partial \tilde{N}_{1-\nu}(\tau_k(\bar{\mathbf{x}}))} H(\tilde{N}_{1-\nu}(\tau_k(\bar{\mathbf{x}})), \mu_{B^k}(\bar{y}^r); \nu) \tag{64}$$

$$\frac{\partial I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r)}{\partial \mu_{B^k}(\bar{y}^r)} = \frac{\partial}{\partial \mu_{B^k}(\bar{y}^r)} H(\tilde{N}_{1-\nu}(\tau_k(\bar{\mathbf{x}})), \mu_{B^k}(\bar{y}^r); \nu). \tag{65}$$

## D. Block of Aggregation

The errors propagated through the block of aggregation (see Fig. 8) are given by

$$\varepsilon_r^{\mathrm{agr}}\{\nu\} = \varepsilon_r^{\mathrm{agr}} \frac{\partial \mathrm{agr}_r(\bar{\mathbf{x}}, \bar{y}^r)}{\partial (1-\nu)} \frac{\partial N(\nu)}{\partial \nu} \tag{66}$$

$$\varepsilon_r^{\mathrm{agr}}\{I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r)\} = \varepsilon_r^{\mathrm{agr}} \frac{\partial \mathrm{agr}_r(\bar{\mathbf{x}}, \bar{y}^r)}{\partial I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r)} \tag{67}$$

where

$$\frac{\partial \mathrm{agr}_r(\bar{\mathbf{x}}, \bar{y}^r)}{\partial (1-\nu)} = \frac{\partial}{\partial (1-\nu)} H(I_{1,r}(\bar{\mathbf{x}}, \bar{y}^r), \ldots, I_{N,r}(\bar{\mathbf{x}}, \bar{y}^r); 1-\nu) \tag{68}$$

$$\frac{\partial \mathrm{agr}_r(\bar{\mathbf{x}}, \bar{y}^r)}{\partial I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r)} = \frac{\partial}{\partial I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r)} H(I_{1,r}(\bar{\mathbf{x}}, \bar{y}^r), \ldots, I_{N,r}(\bar{\mathbf{x}}, \bar{y}^r); 1-\nu). \tag{69}$$

## E. Block of Defuzzification

The errors propagated through the block of defuzzification (see Fig. 9) are given by

$$\varepsilon^{\mathrm{def}}\{\bar{y}^r\} = \varepsilon^{\mathrm{def}} \frac{\partial}{\partial \bar{y}^r} \mathrm{def}\begin{pmatrix} \mathrm{agr}_1(\bar{\mathbf{x}}, \bar{y}^1), \ldots, \mathrm{agr}_N(\bar{\mathbf{x}}, \bar{y}^N) \\ \bar{y}^1, \ldots, \bar{y}^N \end{pmatrix} \tag{70}$$

$$\varepsilon^{\mathrm{def}}\{\mathrm{agr}_r(\bar{\mathbf{x}}, \bar{y}^r)\} = \varepsilon^{\mathrm{def}} \frac{\partial \mathrm{def}\begin{pmatrix} \mathrm{agr}_1(\bar{\mathbf{x}}, \bar{y}^1), \ldots, \mathrm{agr}_N(\bar{\mathbf{x}}, \bar{y}^N) \\ \bar{y}^1, \ldots, \bar{y}^N \end{pmatrix}}{\partial \mathrm{agr}_r(\bar{\mathbf{x}}, \bar{y}^r)}. \tag{71}$$

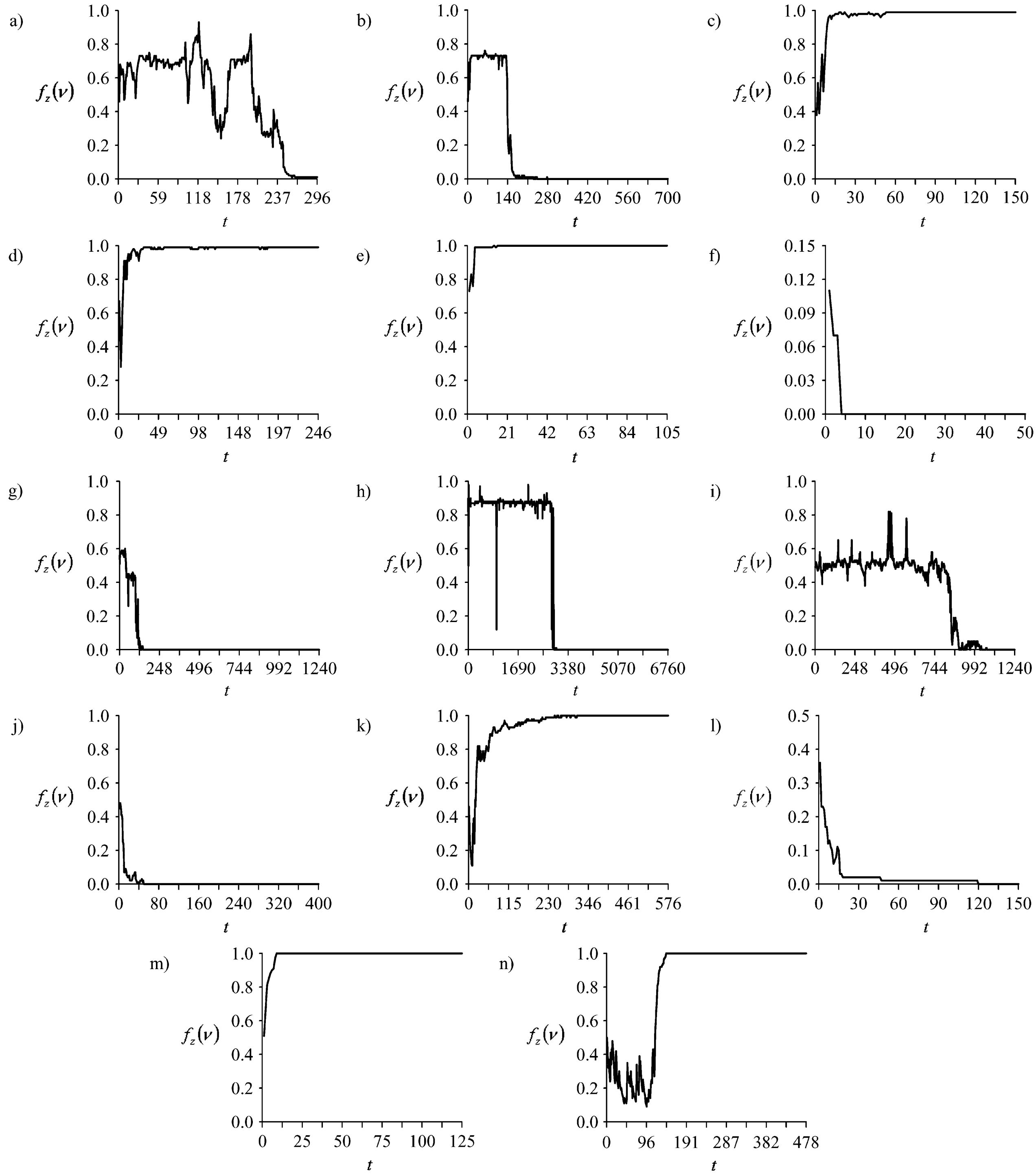


Fig. 10. Illustration of the learning of parameter $\nu$. a) Box and Jenkins Gas Furnace, b) Chemical Plant, c) Glass Identification, d) Ionosphere, e) Iris, f) HANG, g) Monk1, h) Monk2, i) Monk3, j) Nonlinear Dynamic Plant, k) Pima Indians Diabetes, l) Rice Taste, m) Wine Recognition, and n) Wisconsin Breast Cancer.

### *F. Constraint*

Parameter $\nu$ in procedure (37) and (38) should satisfy the condition $0 \leq \nu \leq 1$. Therefore, in the process of learning we replace $\nu$ by $f_z(\nu)$ defined as follows:

$$f_z(\nu) = \frac{1}{1 + \exp(-(p_1\nu - p_2))}. \tag{72}$$

Consequently, the compromise operator (4) takes the form

$$\tilde{N}_\nu(a) = (1 - f_z(\nu))N(a) + f_z(\nu)a. \tag{73}$$

## VI. Simulation Examples

In this paper, we simulate the following examples:

a) Box and Jenkins gas furnace [1];
b) Chemical Plant [25];
c) Glass Identification [26];
d) Ionosphere [26];
e) Iris [26];
f) HANG [25];
g) Monk1 [26];

TABLE II
EXPERIMENTAL RESULTS

| Example number | Name of simulation | Neuro-fuzzy system with an H-implication | | | |
|---|---|---|---|---|---|
| | | Initial values of $\nu$ | Final values of $\nu$ after learning | RMSE or mistakes [%] (learning sequence) | RMSE or mistakes [%] (testing sequence) |
| a) | Box and Jenkins Gas Furnace | 0.5 | 0.0000 | 0.4973 | - |
| b) | Chemical Plant | 0.5 | 0.0000 | 0.0060 | - |
| c) | Glass Identification | 0.5 | 1.0000 | 3.33% | 3.13% |
| d) | Ionosphere | 0.5 | 1.0000 | 1.63% | 8.57% |
| e) | Iris | 0.5 | 1.0000 | 0.95% | 4.44% |
| f) | HANG | 0.5 | 0.0000 | 0.1059 | - |
| g) | Monk1 | 0.5 | 0.0000 | 0.81% | 3.01% |
| h) | Monk2 | 0.5 | 0.0000 | 9.47% | 9.03% |
| i) | Monk3 | 0.5 | 0.0000 | 0.00% | 3.01% |
| j) | Nonlinear Dynamic Plant | 0.5 | 0.0000 | 0.0238 | 0.0133 |
| k) | Pima Indians Diabetes | 0.5 | 1.0000 | 20.80% | 23.51% |
| l) | Rice Taste | 0.5 | 0.0000 | 0.0185 | 0.0545 |
| m) | Wine Recognition | 0.5 | 1.0000 | 0.00% | 1.89% |
| n) | Wisconsin Breast Cancer | 0.5 | 1.0000 | 2.51% | 1.95% |

h) Monk2 [26];
i) Monk3 [26];
j) Nonlinear Dynamic Plant [27];
k) Pima Indians Diabetes [26];
l) Rice Taste [20];
m) Wine Recognition [26];
n) Wisconsin Breast Cancer [26].

Each of the simulations is designed in the same fashion.

i) In the first experiment, based on the input-output data, we learn the parameters of the membership functions and a system type $\nu \in [0,1]$ of the neuro-fuzzy inference system described by formulas (36)–(39). In this experiment the neuro-fuzzy system is based on the H-function generated by the algebraic t-norm (see Examples 1 and 3). It will be seen that the optimal values of $\nu$, determined by a gradient procedure, are either zero or one. The results are depicted in Table II. The learning of parameter $\nu$ is replaced by the learning of function (72). The results are illustrated in Fig. 10 which shows the learning of function (72) versus the number of iterations. It was assumed that $p_1 = 10$ and $p_2 = 5$.
ii) In the second experiment, we learn the parameters of the membership functions of the Mamdani-type system. As the "engineering implication," we apply the product operator. The system is described by (40). The results are shown in Table III.
iii) In the third experiment we learn the parameters of the membership functions of the logical-type system. As the fuzzy implication we apply the Reichenbach operator given by (23). The system is described by (41). The results are presented in Table IV.

TABLE III
EXPERIMENTAL RESULTS

| Example number | Name of simulation | Neuro-fuzzy system of the Mamdani-type | |
|---|---|---|---|
| | | RMSE or mistakes [%] (learning sequence) | RMSE or mistakes [%] (testing sequence) |
| a) | Box and Jenkins Gas Furnace | 0.4912 | - |
| b) | Chemical Plant | 0.0059 | - |
| c) | Glass Identification | 4.67% | 3.13% |
| d) | Ionosphere | 2.03% | 11.43% |
| e) | Iris | 0.95% | 6.67% |
| f) | HANG | 0.1098 | - |
| g) | Monk1 | 0.81% | 3.01% |
| h) | Monk2 | 8.28% | 9.95% |
| i) | Monk3 | 0.00% | 3.24% |
| j) | Nonlinear Dynamic Plant | 0.0241 | 0.0132 |
| k) | Pima Indians Diabetes | 21.60% | 25.37% |
| l) | Rice Taste | 0.0186 | 0.0620 |
| m) | Wine Recognition | 0.80% | 3.77% |
| n) | Wisconsin Breast Cancer | 2.51% | 2.44% |

Analyzing Tables II–IV, we draw the following conclusions.

i) The neuro-fuzzy system (39) becomes of the Mamdani-type $\nu = 0$ for examples a), b), f), g), h), i), j), and l).
ii) The neuro-fuzzy system (39) becomes of the logical-type $\nu = 1$ for examples c), d), e), k), m), and n).
iii) The Mamdani-type systems are more suitable to approximation problems, whereas the logical-type systems may be preferred for classification problems.
iv) Comparing Tables II, III, and IV, we see that when parameter $\nu$ in system (39) takes the final value equal 0 (system

TABLE IV
EXPERIMENTAL RESULTS

| Example number | Name of simulation | Neuro-fuzzy system of the logical-type | |
|---|---|---|---|
| | | RMSE or mistakes [%] (learning sequence) | RMSE or mistakes [%] (testing sequence) |
| a) | Box and Jenkins Gas Furnace | 0.6188 | - |
| b) | Chemical Plant | 0.0066 | - |
| c) | Glass Identification | 3.33% | 3.13% |
| d) | Ionosphere | 1.63% | 9.52% |
| e) | Iris | 0.95% | 4.44% |
| f) | HANG | 0.1205 | - |
| g) | Monk1 | 4.03% | 9.49% |
| h) | Monk2 | 18.34% | 22.92% |
| i) | Monk3 | 1.61% | 5.09% |
| j) | Nonlinear Dynamic Plant | 0.0322 | 0.0209 |
| k) | Pima Indians Diabetes | 20.80% | 23.51% |
| l) | Rice Taste | 0.0198 | 0.0707 |
| m) | Wine Recognition | 0.00% | 1.89% |
| n) | Wisconsin Breast Cancer | 2.51% | 1.95% |

is of a Mamdani-type) then the performance (rmse or mistakes [%]) of the Mamdani-type system given by (40) is better than the performance of the logical-type system given by (41). Similarly, when parameter $\nu$ in system (39) takes the final value equal 1 (system is of a logical-type) then the performance (rmse or mistakes [%]) of the logical-type system given by (41) is better than the performance of the Mamdani-type system given by (40).

v) The influence of simultaneous tuning of parameter $\nu$ and parameters of Gaussian membership functions in system (36)–(39) on the final performance is similar to the influence of tuning only parameters of membership functions in system (40) or (41), provided that the type of the system is correctly chosen. Otherwise, tuning of both $\nu$ and parameters of membership functions gives better results.

vi) Parameter $\nu$ after learning does not take any value in the interval $(0,1)$ because only for $\nu = 0$ or $\nu = 1$ the system described by (36)–(39) is well defined (in terms of a t-norm or t-conorm). It should be noted that function (72) cuts off the learning process when $f_z(\nu) < 0$ or $f_z(\nu) > 1$.

## VII. FINAL REMARKS

In this paper, we proposed a new class of operators called quasi-triangular norms. The operators have been applied to design a new class of neuro-fuzzy systems. At the design stage we do not assume a concrete form of a fuzzy inference. Therefore, we applied the H-operators and trained neuro-fuzzy systems described by (36)–(39). The inference of neuro-fuzzy systems has been established in the process of learning: either Mamdani-type represented by a t-norm or logical-type represented by an S-implication. When the learning process is finished, the trained neuro-fuzzy systems based on H-operators take simpler forms based on t-norms or t-conorms.

## APPENDIX

### A. Zadeh's Operator

$$N(a) = 1 - a \tag{74}$$

$$\frac{\partial N(a)}{\partial a} = -1. \tag{75}$$

### B. Compromise Operator

$$\tilde{N}_\nu(a) = (1 - f_z(\nu))(1-a) + f_z(\nu)a \tag{76}$$

$$\frac{\partial \tilde{N}_\nu(a)}{\partial a} = 2f_z(\nu) - 1 \tag{77}$$

$$\frac{\partial \tilde{N}_\nu(a)}{\partial \nu} = (2a-1)\frac{\partial f_z(\nu)}{\partial \nu}. \tag{78}$$

### C. Defuzzification Operator

$$\begin{aligned}&\mathrm{def}(a_1, a_2, \ldots, a_n; w_1, w_2, \ldots, w_n)\\ &\quad = \mathrm{def}(\mathbf{a};\mathbf{w}) = \frac{\sum_{i=1}^n w_i a_i}{\sum_{i=1}^n a_i}\end{aligned} \tag{79}$$

$$\begin{aligned}&\frac{\partial \mathrm{def}(\mathbf{a};\mathbf{w})}{\partial a_j}\\ &\quad = (w_j - \mathrm{def}(\mathbf{a};\mathbf{w}))\frac{1}{\sum_{i=1}^n a_i}\end{aligned} \tag{80}$$

$$\begin{aligned}&\frac{\partial \mathrm{def}(\mathbf{a};\mathbf{w})}{\partial w_j}\\ &\quad = \left(a_j - \mathrm{def}(\mathbf{a};\mathbf{w})\frac{\partial a_j}{\partial w_j}\right)\frac{1}{\sum_{i=1}^n a_i}.\end{aligned} \tag{81}$$

### D. Gaussian Membership Function

$$\mu_A(x) = \exp\left(-\left(\frac{x-\bar{x}}{\sigma}\right)^2\right) \tag{82}$$

$$\frac{\partial \mu_A(x)}{\partial x} = -\mu_A(x)\frac{2(x-\bar{x})}{\sigma^2} \tag{83}$$

$$\frac{\partial \mu_A(x)}{\partial \bar{x}} = \mu_A(x)\frac{2(x-\bar{x})}{\sigma^2} \tag{84}$$

$$\frac{\partial \mu_A(x)}{\partial \sigma} = \mu_A(x)\frac{2(x-\bar{x})^2}{\sigma^3}. \tag{85}$$

### *E. Constraint on* $\nu$

$$f_z(x) = \frac{1}{1+\exp(-(p_1 x - p_2))} \tag{86}$$

where $p_1 = 10$ and $p_2 = 5$

$$\frac{\partial f_z(x)}{\partial x} = p_1(1 - f_z(x))f_z(x). \tag{87}$$

### *F. H-Function Generated by the Product t-Norm*

$$H(\mathbf{a};\nu) = \tilde{N}_\nu\left(1 - \prod_{i=1}^{n}(1 - \tilde{N}_\nu(a_i))\right) \tag{88}$$

$$H(\mathbf{a},\nu) = \tilde{N}_\nu(h(\mathbf{a};\nu)) \tag{89}$$

where

$$h(\mathbf{a};\nu) = 1 - \prod_{i=1}^{n}(1 - \tilde{N}_\nu(a_i)) \tag{90}$$

$$\frac{\partial H(\mathbf{a};\nu)}{\partial \nu} = \left((2h(\mathbf{a};\nu) - 1) + (2f_z(\nu) - 1) \times \sum_{i=1}^{n}\left(\frac{(2a_i - 1)}{\prod_{\substack{u=1\\u\neq i}}^{n}(1 - \tilde{N}_\nu(a_u))}\right)\right)\frac{\partial f_z(\nu)}{\partial \nu}. \tag{91}$$

$$\frac{\partial H(\mathbf{a};\nu)}{\partial a_j} = (2f_z(v) - 1)^2 \prod_{\substack{u=1\\u\neq j}}^{n}(1 - \tilde{N}_\nu(a_u)) \tag{92}$$

## ACKNOWLEDGMENT

The authors would like to thank the referees for their helpful suggestions and comments.

## REFERENCES

[1] G. E. P. Box and G. M. Jenkins, *Time Series Analysis, Forecasting and Control*. San Francisco, CA: Holden Day, 1970.
[2] T. Calvo and R. Mesiar, "Continuous generated associative aggregation operators," *Fuzzy Sets Syst.*, vol. 126, pp. 191–199, 2002.
[3] O. Cordon, F. Herrera, and A. Peregrin, "Applicability of the fuzzy operators in the design of fuzzy logic controllers," *Fuzzy Sets Syst.*, vol. 86, pp. 15–41, 1997.
[4] E. Czogala and J. Łęski, *Fuzzy and Neuro-Fuzzy Intelligent Systems*. New York: Physica-Verlag, 2000.
[5] J. C. Fodor, "On fuzzy implication operators," *Fuzzy Sets Syst.*, vol. 42, pp. 293–300, 1991.
[6] J. C. Fodor, R. R. Yager, and A. Rybalov, "Structure of uninorms," *Int. J. Uncertainty Fuzziness Knowledge-Based Syst.*, vol. 5, no. 4, pp. 411–427, 1997.
[7] M. J. Frank, "On the simultaneous associativity of $F(x,y)$ and $x + y - F(x,y)$," *Aequationes Math.*, vol. 19, pp. 194–226, 1979.
[8] M. Gorzałczany, *Computational Intelligence Systems and Applications: Neuro-Fuzzy and Fuzzy Neural Synergisms*. New York: Springer-Verlag, 2002.
[9] J. S. R. Jang and C. T. Sun, "Neuro-fuzzy modeling and control," *Proc. IEEE*, vol. 83, no. 3, pp. 378–406, Mar. 1995.
[10] J. S. Jang, C. T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*. Upper Saddle River, NJ: Prentice-Hall, 1997.
[11] S. Jensei, "A note on the ordinal sum theorem and its consequence for the construction of triangular norms," *Fuzzy Sets Syst.*, vol. 126, pp. 199–207, 2002.
[12] C. F. Juang and C. T. Lin, "An online self-constructing neural fuzzy inference network and its applications," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 1, pp. 12–32, Feb. 1998.
[13] N. Kasabov, *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*. Cambridge, MA: MIT Press, 1996.
[14] E. P. Klement, R. Mesiar, and E. Pap, *Triangular Norms*. Dordrecht, Netherlands: Kluwer, 2000.
[15] C. T. Lin and G. C. S. Lee, *Neural Fuzzy Systems-A Neuro-Fuzzy Synergism to Intelligent Systems*. Upper Saddle River, NJ: Prentice-Hall, 1997.
[16] M. Mas, G. Mayor, and J. Torrens, "The modularity condition for uninorms and t-operators," *Fuzzy Sets Syst.*, vol. 126, pp. 207–219, 2002.
[17] J. M. Mendel, "Fuzzy logic systems for engineering: A tutorial," *Proc. IEEE*, vol. 83, no. 3, pp. 345–377, Mar. 1995.
[18] ——, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Upper Saddle River, NJ: Prentice-Hall , 2001.
[19] D. Nauck, F. Klawon, and R. Kruse, *Foundations of Neuro-Fuzzy Systems*, Chichester, U.K.: Wiley, 1997.
[20] K. Nozaki, H. Ishibuchi, and K. Tanaka, "A simple but powerful heuristic method for generating fuzzy rules from numerical data," *Fuzzy Sets Syst.*, vol. 86, pp. 251–270, 1995.
[21] W. Pedrycz and F. Gomide, *An Introduction to Fuzzy Sets*. Cambridge, MA: MIT Press, 1998.
[22] D. Rutkowska, *Neuro-Fuzzy Architectures and Hybrid Learning*. New York: Springer-Verlag, 2002.
[23] L. Rutkowski and K. Cpałka, "A general approach to neuro-fuzzy systems," presented at the 10th IEEE Int. Conf. Fuzzy Systems, Melbourne, Australia, 2001.
[24] ——, *Flexible Structures of Neuro-Fuzzy Systems, Quo Vadis Computational Intelligence, Studies in Fuzziness and Soft Computing*. New York: Springer-Verlag, 2000, vol. 54, pp. 479–484.
[25] M. Sugeno and T. Yasukawa, "A fuzzy-logic based approach to qualitative modeling," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 1, pp. 7–31, Feb. 1993.
[26] UCI Respository of Machine Learning Databases, C. J. Mertz and P. M. Murphy. [Online]. Available: http://www.ics.uci.edu/pub/machine-learning-databases
[27] L. Wang and J. Yen, "Application of statistical information criteria for optimal fuzzy model construction," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 4, pp. 362–371, Aug. 1998.
[28] R. R. Yager, *Fuzzy Sets, Neural Networks and Soft Computing*. New York: Van Nostrand Reinhold, 1997.
[29] R. R. Yager and D. P. Filev, *Essentials of Fuzzy Modeling and Control*. New York: Wiley, 1994.

**Leszek Rutkowski** (M'92–SM'94–F'02) was born in Wrocław, Poland, in 1952. He received the M.Sc., Ph.D., and D.Sc. degrees in 1977, 1980, 1986, respectively, all from the Technical University of Wrocław.

Since 1980, he has been with the Technical University of Częstochowa, Częstochowa, Poland, where he is currently a Professor and Chairman of the Computer Engineering Department. From 1987 to 1990, he held a visiting position with the School of Electrical and Computer Engineering at Oklahoma State University, Stillwater. His research interests include neural networks, fuzzy systems, computational intelligence, pattern recognition, and systems identification. He published over 100 technical papers, including 17 in various IEEE TRANSACTIONS. He is the author/coauthor of numerous books. He is also President and Founder of the Polish Neural Networks Society. He organized and served as General Chair of the Polish Neural Networks Conferences held in 1996, 1997, 1999, 2000, 2002, and 2004.

Dr. Rutkowski is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS.

**Krzysztof Cpałka** was born in Częstochowa, Poland, in 1972. He received the M.Sc. degree in electrical engineering and the Ph.D. degree (Honors) in computer engineering, both from the Technical University of Częstochowa, in 1997 and 2002, respectively.

Since 2002, he has been an Assistant Professor in the Department of Computer Engineering, the Technical University of Częstochowa. His research interests include fuzzy systems, neural networks, evolutionary algorithms, and artificial intelligence. He published over 30 technical papers in journals and conference proceedings.