1 On functional equivalence of certain fuzzy controllers and RBF type approximation schemes *

László T. Kóczy[1,2], Domonkos Tikk[1,2], and Tamás D. Gedeon[2]

[1] Dept. of Telecom. & Telematics
Budapest University of Technology and Economics
H-1117 Budapest, Pázmány sétány 1/d, Hungary
[2] School of Information Technology, Murdoch University
South Street, Murdoch, 6150 W.A., Australia

Abstract. Both general fuzzy systems and most neural networks are universal approximators in the sense that they are capable of approximating any continuous function with arbitrary accuracy with respect to, e.g., the supremum norm. It means that these techniques share approximation capabilities. However, the way they captures the underlying transfer function is different. Fuzzy systems operating with if-then rules have the advantage of easy linguistic interpretability, while neural networks can adapt learning methods to improve their performance according to a training data set. We point out in this paper that several fuzzy controllers implement one of the typical neural networks (having radial basis type activation functions), and hence, their combination may alloy the the advantageous properties of the two techniques.

1.1 Introduction

One of the most important and intriguing goals of Artificial Intelligence is to construct methods, models and algorithms to cope with very complex systems, often lacking known analytical models. Such approaches have attempted to follow examples of natural intelligence and biological organisms. The classical AI approach was based on symbolic logic and its tautologies, such as *Modus Ponens, Hypothetical Syllogism*, etc. The former is the simple logical scheme that works when an implication and its antecedent part are known to be true; then, its consequent part is concluded to also be true:

$$\begin{array}{l} A \to B \\ A \\ \hline B \end{array}$$

A great disadvantage of this approach is that knowledge stored in the form of implications does not relate directly to the structure of the state

* This research was supported by the Australian Research Council, the Hungarian Ministry of Education (MKM) FKFP 0235/1997 and FKFP 0422/1997 and the National Science Research Fund (OTKA) 019671 and 030655.

space of the problem in question. Most physical phenomena can be modeled in a state space where each dimension is a subset of $\mathbb{R}^n$, usually a hyper interval $X = X_1 \times X_2 \times \cdots \times X_n = \prod_{i=1}^{n}[L_i, U_i]$ where L_i (U_i) denotes the lower (upper) bound of the interval in the ith dimension. ($\prod$ denoting Cartesian product). Consequently, there is an ordering $\prec_i \subset X_i \times X_i$ ($i = 1, \ldots, n$) in every dimension of X, and a partial ordering $\prec$ in X itself, so that $\underline{x}_1 \prec \underline{x}_2 \Leftrightarrow x_{i1} \prec_i x_{i2}$ for all $i = 1, \ldots, n$. (Throughout this work, as our denotation convention, we will use n for the dimension of the input space, $\underline{x}$, and $\underline{\underline{A}}$ for vector x and matrix A, respectively.) Further, there is a norm $\|\cdot\|$ in X, usually the Euclidean norm. When applying a symbolic logical approach to a system in the state space X, it is necessary to quantify this space, so that every "quantum" or interval is denoted by a symbol (see Figure 1.1). The symbols A_{i_k} represent the whole, kth, interval in X_i, and there is no way to differentiate among values within the same quantum or interval. This fact results in a certain coarseness of the model that can be refined by applying smaller intervals and more symbols. It is a more significant problem, however, that the relative position of these symbols to each other cannot be expressed at this level, e.g. the fact that $A_{i_k} \prec_i A_{i_{k+1}} \prec_i A_{i_{k+2}}$ (i.e. in the sense that the typical values or centres are ordered according to $\prec_i$). The same applies to relations like $\|A_{i_k} - A_{i_{k+1}}\| < \|A_{i_k} - A_{i_{k+2}}\|$, as the norm and the distance between two logical symbols are senseless without the additional information referring to the metrics of the space ("$<$" meaning here the usual ordering on real numbers).

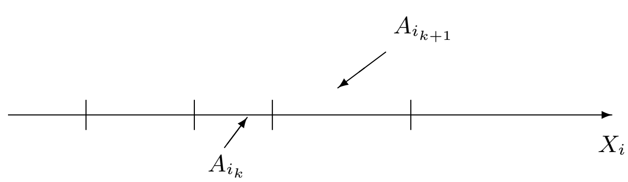

Fig. 1.1. Quantification and symbol correspondence

A knowledge base consisting of symbolic rules has the form

$$R = \{R_j | j = 1, \ldots, r\}, \quad R_j = A_j \rightarrow B_j$$

(r is the number of the rules), where $A_j = \prod_{i=1}^{n} A_{ji}$, and each A_{ji} is a logic symbol in X_i (the antecedent of the jth rule in the ith dimension). A fact or observation in such a system has the form $A^* = \prod_{i=1}^{n} A_i^*$, and it must match exactly with the antecedent in one of the R_j ($j = 1, \ldots, r$) in order to make it possible to apply e.g. the *Modus Ponens* in order to obtain a conclusion (necessarily equal to one of the rule consequents).

It was an essential step towards establishing a means for describing the internal structure of the state space, when the use of fuzzy symbols was proposed [26]. By these, based on the information present in the membership functions attached to each symbol, the intervals close to each other were merging into single "fuzzy intervals" or "fuzzy numbers" By the possibility of partially overlapping membership functions the partial semantic overlapping of the meaning of "close by" symbols could be formally expressed (see Figure 1.2).

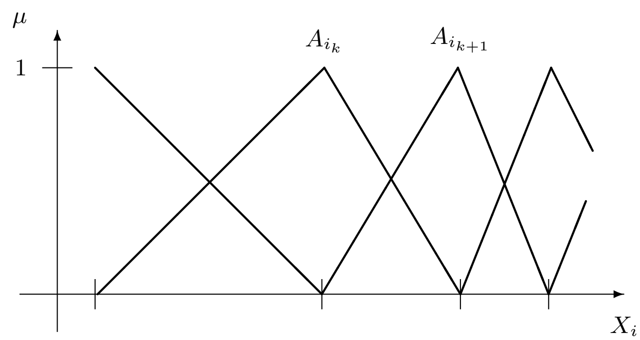

Fig. 1.2. Symbols with fuzzy membership functions

A similar feature is inherent for artificial neural networks, where sub-symbolic information is contained in the activation functions and weights. It is obvious that sub-symbolic models have a smaller number of symbols, and consequently, a smaller number of rules in the base, although in this case each symbol has a much more complicated description in the form of an additional membership function. Let us assume that there are T_i symbols used in dimension X_i. Then the number of rules in the base is $|R| = \prod_{i=1}^{n} T_i$. Let $T_i \leq T$ for all $i = 1, \ldots, n$, then the number of rules in the base is at most $|R| \leq T^n$. Let c be the rate of reduction of the number of symbols when fuzzy membership functions are introduced, and let e be the rate of extension of the amount of information belonging to a single symbol. Then, instead of approximately T^n pieces of information, only $\left(\frac{T}{c}\right)^n e$ will be needed. However, $\left(\frac{T}{c}\right)^n e < T^n$ holds, whenever $c > e^{\frac{1}{n}}$, i.e. when n is large enough ($n > \frac{\log e}{\log c}$). Obviously, the introduction of fuzzy symbols leads to the reduction of computational complexity when the number of state space dimensions is large enough.

The reason for this effective reduction is that the use of fuzzy membership functions utilizes the knowledge about the structure of the space and introduces a kind of interpolation.

Calculations about the explicit interpolation functions with triangular and trapezoidal membership functions in various fuzzy reasoning and control systems can be found in [13].

A common feature of such fuzzy rule bases is that the antecedents form a collection of fuzzy sets, where their kernels (either single points or intervals) represent typical values of $\underline{x}$, and between these kernels the membership functions cover the space, so that for every $\underline{x} \in X$ there is at least one such antecedent, which is true over $\underline{x}$ at least to a fixed degree $\alpha > 0$. (Usually $\alpha \geq 0.5$.) If this latter condition is satisfied, it is true that there is at least one rule for every x, which is at least as true as it is false. (If $\alpha > 0.5$ is satisfied then there is at least one rule which is more true than false.) If

$$\forall x_i \in X_i,\ i = 1, \ldots, n\ \exists j\ (j \in \{1, \ldots, r\}) : A_{ji}(x_i) \geq \alpha > 0$$

then $\{A_{ji} | j = 1, \ldots, r\}$ form an α-cover of X_i. If also

$$\forall \underline{x} \in X,\ \exists j\ (j \in \{1, \ldots, r\}) : A_j(\underline{x}) \geq \alpha > 0$$

is fulfilled, then $\{A_j | j = 1, \ldots, r\}$ is an α-cover of X. In the original Mamdani-algorithm [15], e.g. the antecedents formed a cover for approximately $\alpha = 0.75$. In most current applications however $\alpha = 0.5$ is satisfactory.

If $R = \{R_j | j = 1, \ldots, r\}$ is a *full set of rules*, then for arbitrarily chosen n rules rules $R_{j_1}, R_{j_2}, \ldots, R_{j_n} \in R$, ($A_{j_\ell,i}$ being the ith projection of A_{j_ℓ}, the antecedent of rule R_{j_ℓ}, to the sub-space X_i $(i, \ell = 1, \ldots, n)$)

$$\exists m \in \{1, \ldots, r\} : A_m = A_{j_1,1} \times A_{j_2,1} \times \cdots \times A_{j_n,n}.$$

holds. This means that the rule antecedents form a "grid" in X as illustrated in Figure 1.3. Full rule bases contain $|R| = \prod_{i=1}^{n} r_i$ rules, where r_i stands for the number of different antecedent projections appearing in dimension X_i. It is reasonable to use here k subscripts, where the ith one indicates that the antecedent of the given rule generates the ith projection in X_i if all projections are ordered along each component axis. In most practical applications the rule set is full, unless some rules in the original set are omitted in order to decrease the size of the rule base.

In most cases the antecedents also form a *Ruspini-partition* of X, meaning that

$$\forall \underline{x} \in X : \sum_{j=1}^{r} A_j(\underline{x}) = 1$$

Full rule bases with the property of Ruspini-partition are usually 0.5-covers.

1.2 Fuzzy rule based controllers

It is obvious that although the introduction of fuzzy membership functions for describing the sub-symbolic information behind the symbols leads to the reduction of the overall computational complexity, whenever k is large enough the reduction is not radical [11]. There are further possibilities to decrease the effective value of T, by applying rule bases where the antecedents do not form an α-cover of the input state space.

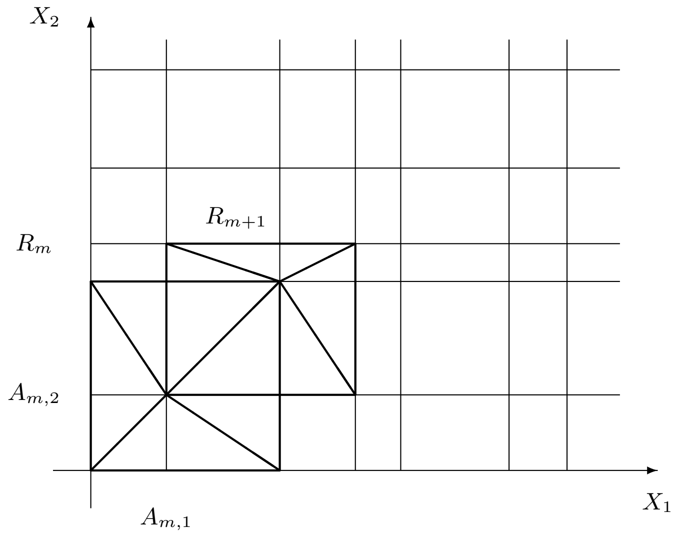

Fig. 1.3. Rules represented by their supports in a full rule base in two-dimensional space

The original idea of reasoning and control within fuzzy rule bases was proposed by Zadeh [27], called the Compositional Rule of Inference (CRI) and had the disadvantage of running directly in the k-dimensional input variable space, while being able to describe multi-dimensional membership function distributions of arbitrary shape. Its modified version, the Mamdani-algorithm [15] applied the projections of the antecedents and thus returned to the form of rules $R_j = A_j \to B_j$, where $A_j = \prod_{i=1}^{n} A_{ji}$, meaning that only such A_j could be used that were the cylindrical closures of some membership functions A_{ji}, each being of the type $A_{ji} : X_i \to [0, 1]$. This method offers much better computational speed.

The idea of the Mamdani-algorithm is to use the heights of the intersections of the observation with the antecedents and weighing the consequents with these, e.g. by simply "cutting them" at the given height:

$$B^* = \max_j \left\{ \min \left\{ \text{height}(A^*(\underline{x}), A_j(\underline{x})), B_j(y) \right\} \right\}, \tag{1.1}$$

where $\text{height}(S(\underline{x})) = \max_{\underline{x}} S(\underline{x})$. In the Larsen-method, the modification is that the consequents are multiplied with these heights:

$$B^* = \max_j \left\{ \text{height}(A^*(\underline{x}), A_j(\underline{x})) B_j(y) \right\}.$$

In both cases, y^* is obtained by defuzzification, e.g. in the most popular techniques, calculating the centre of gravity of the modified consequents. This latter means that instead of taking the maximum, in effect the sum of the membership functions will be considered:

$$B^* = \sum_{j=1}^{r} \text{height}(A^*(\underline{x}), A_j(\underline{x}))B_j(y),$$

which is in case of a crisp singleton observation identical with

$$B^*_{\text{eff}} = \sum_{j=1}^{r} A_j(\underline{x}^*)B_j(y).$$

This latter is the most popular formula used for fuzzy control applications.

1.3 Interpolation in sparse fuzzy rule bases

If for given α the antecedents form no α-cover, the rule base is α-sparse. If they do not form a cover for any $\alpha > 0$, the rule base is sparse. The CRI, the Mamdani method and its variants, the Takagi–Sugeno method [21], which applies linear functions of $\underline{x}$ in the consequent part, and all other fuzzy control approaches based on some form of degree of matching between observation and antecedents cannot be used for sparse rule systems.

Instead, a more general notion of the degree of similarity was introduced, based on the family of *α-distances between two convex and normal fuzzy sets* (CNF sets), which is defined as

$$\begin{aligned}\widetilde{d}(A_1, A_2) &= \{d_{\alpha,C}(A_1, A_2),\ \alpha \in (0, 1], C \in \{L, U\}\} \\ &= \{\|A_{1\alpha L} - A_{2\alpha L}\|, \|A_{1\alpha U} - A_{2\alpha U}\|, \alpha \in (0, 1]\}\end{aligned} \tag{1.2}$$

subscripts L and U denoting the minimum and maximum of the respective α-cuts. For simplicity, the distance belonging to fixed and L or U will be denoted by $d_{\alpha,C}(A_1, A_2)$. The conditions for the existence of the fuzzy distance set is that both fuzzy sets are CNF, and that they are comparable in the sense of $\prec$.

Using this notion of distance (1.2), the fuzzy similarity set can be defined as

$$\begin{aligned}\widetilde{s}(A_1, A_2) &= \{s_{\alpha,C}(A_1, A_2),\ \alpha \in (0, 1], C \in \{L, U\}\} \\ &= \left\{\frac{1}{d_{\alpha,C}(A_1, A_2)}, \alpha \in (0, 1]\right\}\end{aligned} \tag{1.3}$$

the elements of the similarity degree set being the reciprocals of the elements of the distance set.

The basic idea of the rule interpolation is formulated in the *Fundamental Equation of Rule Interpolation* (FERI):

$$D(A^*, A_1) : D(A^*, A_2) = D(B^*, B_1) : D(B^*, B_2). \tag{1.4}$$

In this equation A^* and B^* denote the observation and the corresponding conclusion, while $R_1 = A_1 \to B_1$, $R_2 = A_2 \to B_2$ are the rules to be interpolated, such that $A_1 \prec A^* \prec A_2$ and $B_1 \prec B_2$. The FERI states that the ratio of the distances in the input space or spaces (between the antecedents and the observation) should be equal to the corresponding ratio of the distances in the output space (between the proper rule consequents and the conclusion). If D denotes the Euclidean distance between two symbols, the solution for B^* results in simple linear interpolation. If $D = \widetilde{d}$ (the fuzzy distance family), linear interpolation between corresponding α-cuts is performed.

A more general form of (1.4) gives

$$B^*_\alpha = \sum_{j=1}^{r} s_\alpha(A^*, A_j) B^*_{j\alpha}, \quad \alpha \in (0,1]$$

where s_α is some α-cut related similarity degree, e.g., the fuzzy similarity (1.3) obtained from the reciprocal distances of the α-cuts. This similarity can be considered as an extended "degree of matching", and these similarity degrees replace the degrees $A_j(\underline{x}^*)$ or height$(A_j \wedge A^*)$ ($\wedge$ denoting the appropriate t-norms) which are used in the Mamdani, Larsen, (Takagi–) Sugeno and other classical fuzzy reasoning algorithms. These methods can be substituted by fuzzy interpolation (KH interpolation [12]), which also functions in sparse rule bases. The Mamdani type algorithms are thus replaced by

$$B^*_\alpha = \frac{\sum_{j=1}^{r} s_\alpha(A^*, A_j) B_{j\alpha C}}{\sum_{j=1}^{r} s_\alpha(A^*, A_j)} = \frac{\sum_{j=1}^{r} \frac{1}{d_{\alpha,C}(A^*,A_j)} B_{j\alpha C}}{\sum_{j=1}^{r} \frac{1}{d_{\alpha,C}(A^*,A_j)}} \tag{1.5}$$

where the normalized degree of similarity for fixed α and C is the reciprocal distance of the observation from the corresponding antecedent, divided by the sum of all these distances. It must be noted, however, that B^* reconstructed directly from the above α-cuts does not always exist, as various abnormalities in the shape of the conclusion might necessitate some transformations, which eventually result in obtaining sub-normal conclusions (cf. [12]). It should be noted that in practical applications, it is enough to do calculations for $\alpha \in \mathcal{B}$, the breakpoint set of the membership functions, which is four points altogether [20] in the case of the most widely applied trapezoidal functions $\mathcal{B} = \{0, 1; L, U\}$.

In all these approaches, the approximated mapping is "fuzzy set to fuzzy set":

$$\widetilde{\Psi} : \widetilde{P}(X) \to \widetilde{P}(Y)$$

where $\widetilde{P}(Z)$ denotes the fuzzy power set of Z, i.e. all fuzzy sets of the universe of discourse Z, so that a fuzzy observation is always mapped into a fuzzy conclusion by the respective rule base and inference engine: $\widetilde{\Psi}(A^*) = B^*$. In the special case where the observation is a crisp singleton, we have $\widetilde{\Psi}(\underline{x}^*) = B^*$. In control applications the conclusion set should always be defuzzified, so that $y^* = \text{defuzz}\,(B^*)$. With both crisp singleton inputs and outputs, we get $y^* = \text{defuzz}\left(\widetilde{\Psi}(\underline{x}^*)\right) = \Psi(\underline{x}^*)$. (Some of the most important Ψ-s are explicitly given in [13].) In this sense, fuzzy rule based controllers can be considered as multivariate real function approximators. (For an illustrative example of approximation see the Appendix).

The respective interpolative extension for the equation for the Takagi–Sugeno algorithm is

$$y^* = \frac{\sum_{j=1}^{r} s_\alpha(A^*, A_j)(b_{j_1}\underline{x}^* + b_{j_0})}{\sum_{j=1}^{r} s_\alpha(A^*, A_j)}$$

where $\forall j \in \{1, r\} : b_{j_1} = 0$ in the case of the Sugeno-controller. Here, the similarity degree is an overall one that can be derived from the reciprocal of an overall distance of the two fuzzy sets in question, e.g. by defining it as

$$s(A^*, A_j) = \frac{1}{\int\limits_{\alpha_L \in (0,1] \wedge \alpha_u \in (0,1]} d_{\alpha,C}(A^*, A_j)\mathrm{d}\alpha}. \tag{1.6}$$

In the case of trapezoidal membership functions (1.6) can be simplified to

$$s(A^*, A_j) = \frac{4}{d_{0,L}(A^*, A_j) + d_{1,L}(A^*, A_j) + d_{0,U}(A^*, A_j) + d_{1,U}(A^*, A_j)}.$$

because both the left and right pairs of flanks of the antecedents define trapezoidal areas. This latter interpolation has the advantage of obtaining directly defuzzified consequents, which fact eliminates the problem of the abnormal conclusion shape. This interpolation method will be investigated in more detail in the future.

The investigation of the KH interpolation has shown that using a more radical similarity degree improves the convergence properties tremendously. If s is defined as

$$s_{\alpha,C}(A^*, A_j) = \frac{\frac{1}{d^n_{\alpha,C}(A^*, A_j)}}{\sum_{j=1}^{r} \frac{1}{d^n_{\alpha,C}(A^*, A_j)}}, \tag{1.7}$$

it will be guaranteed that when the consequent membership functions are known ε with accuracy, the conclusion function will be also calculable with $\delta(\varepsilon)$ accuracy [23]. Due to this result, the input-output function of the general KH interpolation obtained by means of the radical similarity defined in (1.7),

$$B^*_{\alpha,C} = \frac{\sum_{j=1}^{r} \frac{1}{d^n_{\alpha,C}(A^*, A_j)} B_{j\alpha C}}{\sum_{j=1}^{r} \frac{1}{d^n_{\alpha,C}(A^*, A_j)}} \tag{1.8}$$

possesses the universal approximation property for every fixed α, i.e., every continuous function defined on a compact set can be approximated arbitrarily accurately with respect to the norm L_p $(p \in \{[1, \infty)\} \cup \infty)$ with functions of this form. By this, the mathematical stability of the method can be formulated simply as follows: If the antecedents or the observation of a rule base change only slightly, it does not cause a significant change in the conclusion.

General KH interpolation has strong connections with the so-called Balázs–Shepard interpolator operator well known in approximation theory. This operator is also capable of approximating real valued continuous functions arbitrarily well, where the convergence of the approximation depends on the modulus of continuity (a certain smoothness factor) of the approximated real function. It has been shown that for every fixed value of α, the function generated by the general KH interpolation is a Balázs–Shepard operator. On the other hand, for practically important piecewise linear membership functions it can be proved that it is sufficient to calculate the conclusion for the elements of the breakpoint set of the membership functions, and in the regions between the breakpoint levels the conclusion can be approximated well by linear functions. Using these results, general KH interpolation can be considered as a generalized function approximator able to approximate fuzzy-to-fuzzy mappings on the Cartesian product of compact universes of discourse.

There are further methods for rule interpolation that we do not discuss here in detail (cf. [2,6,22]).

1.4 Radial Basis Functions in neural networks

1.4.1 RBF approximation scheme

An alternative approach where sub-symbolic information is added is the artificial neural network model. Three-layer feedforward neural networks using *radial basis functions*(RBF) are function approximators. Park and Sandberg proved that any continuous function f could be arbitrarily well approximated on a compact subset $X \subset \mathbb{R}^n$ in L_∞ norm [18]:

$$f(\underline{x}) = \sum_{i=1}^{r} \lambda_i \varphi(\|\underline{x} - \underline{\xi}_i\|) \tag{1.9}$$

where $\varphi : \mathbb{R}^+ \to \mathbb{R}$, $\underline{\xi}_i$ are called the centres of the functions and are real numbers. If there are r known points of the function to be approximated, $\{y_i = F(\underline{x}_i),\ i = 1, \ldots, r\}$, by substituting them into (1.9), a set of linear equations for is obtained [8]:

$$\begin{pmatrix} y_1 \\ \vdots \\ y_r \end{pmatrix} = \begin{pmatrix} \varphi_{11} & \ldots & \varphi_{1r} \\ \vdots & \ddots & \vdots \\ \varphi_{r1} & \ldots & \varphi_{rr} \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_r \end{pmatrix} \tag{1.10}$$

where

$$\varphi_{ij} = \varphi(\|\underline{x}_i - \underline{\xi}_j\|)$$

By denoting $\underline{\underline{\Phi}} = \begin{pmatrix} \varphi_{11} & \dots & \varphi_{1r} \\ \vdots & \ddots & \vdots \\ \varphi_{r1} & \dots & \varphi_{rr} \end{pmatrix}$, $\underline{\lambda} = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_r \end{pmatrix}$, and $\underline{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_r \end{pmatrix}$, (1.10) is simply written as

$$\underline{y} = \underline{\underline{\Phi}}\underline{\lambda}$$

From here, the solutions for λ_i can be obtained by

$$\underline{\lambda} = \underline{\underline{\Phi}}^{-1}\underline{y}$$

if $\underline{\underline{\Phi}}$ is regular. For a rather large set of various functions it has been proved that $\underline{\underline{\Phi}}$ is always regular (see e.g. [5]), if $\underline{\xi}_i$ $(i = 1, \dots, r)$ are distinct vectors. For example, x, $x^2 \log x$, x^3, $(x^2 + c)^\alpha$, $(x^2 + c)^{-1/2}$, and e^{-cx^2} $(c > 0, 0 < \alpha < 1)$. These functions belong to two different groups: *localized* functions which satisfy $\varphi(\underline{z}) \to 0$, when $z \to 0$, and so $\underline{\underline{\Phi}}$ is positive definite, while *non-local* functions $\varphi(z)$ become unbounded, when $z \to 0$, and so $\underline{\underline{\Phi}}$ is not positive definite, having $r - 1$ negative eigenvalues and only a single positive eigenvalue [16].

The RBF approach can be extended to multiple output mappings, and more general approximation, where the number of known points is rather larger than the number of functions φ_i, using the Moore–Penrose pseudo inverse (cf. [8,3]):

$$\underline{\underline{\Phi}}^+ = (\underline{\underline{\Phi}}^T \underline{\underline{\Phi}})^{-1} \underline{\underline{\Phi}}^T,$$

where $\underline{\underline{\Phi}}^+ \underline{\underline{\Phi}} = I_r$, the $r \times r$ identity matrix.

1.4.2 HyperBF approximation with weighted norm

The usually Euclidean norm function used in the RBF approximation (1.9) is suitable when the elements of the vectors to calculate with belong to the same classes. However, when this is not the case, it is more appropriate to consider a general *weighted norm*, which can be defined by its square as [19]:

$$\|\underline{x}\|_C^2 = (\underline{\underline{C}}\underline{x})^T (\underline{\underline{C}}\underline{x}) = \underline{x}^T \underline{\underline{C}}^T \underline{\underline{C}}\underline{x} \tag{1.11}$$

where $\underline{\underline{C}}$ is the *norm weighting matrix* with size $n \times n$, n being the dimension of $\underline{x}$. in the simple case when $\underline{\underline{C}}$ is diagonal, the diagonal elements c_i assign a specific weight to each variable, determining in fact the units of measure and the importance of each input coordinate.

When membership functions are Gaussian

$$\underline{\underline{C}}^T \underline{\underline{C}} = \frac{1}{2} \underline{\underline{\Sigma}}^{-1},$$

where $\underline{\underline{\Sigma}}$ is the *covariance matrix* of a multivariate Gaussian distribution of vector $\underline{x}$. The covariance matrix determines the *receptive field* of the Gaussian radial basis function $\varphi(\|\underline{x} - \underline{\xi}_i\|_C)$. The receptive field is a particular subset of the input domain for which the RBF takes larger value than a prescribed threshold.

We can generalize the RBF approximation scheme by using the weighted norm defined in (1.11) as

$$f(\underline{x}) = \sum_{i=1}^{r} \lambda_i \varphi(\|\underline{x} - \underline{\xi}_i\|_C) \tag{1.12}$$

which scheme is called the *generalized radial*, or *hyper basis function* approximation scheme [19].

The use of the weighted norm can be interpreted as a simple *affine transformation* of the input space. As a special case, the approximation (1.12) contains the original RBF scheme (1.9), when $\underline{\underline{C}} = I_n$, the n dimensional identity matrix. We will return to HyperBF approximation scheme in section 1.5.

1.4.3 RBF networks

Three-layer feedforward neural networks using radial basis functions and with fully interconnected layers are also suitable for approximate continuous functions as the fuzzy systems discussed in section 1.2. The number of units in the input layer should be n, the number of input variables. Each connection between the input layer and the hidden layer should be assigned a value ξ_{ij}, where $i \in \{1, \ldots, r\}$ indicates the subscript of the corresponding unit centre, and $j \in \{1, \ldots, n\}$ refers to the respective component. (In the basic case $r = n$.) The ith hidden unit computes the norm $\|\underline{x}^* - \underline{\xi}_i\|$, and its mapping by φ. The output layer consists of a single unit in the one output variable case, and this unit is connected to the ith hidden unit by a weight λ_i, and is linear, so it generates the value $f(\underline{x}^*) = \sum_{i=1}^{r} \varphi(\|\underline{x}^* - \underline{\xi}_i\|)$ (see Figure 1.4). Multiple output variable cases can be implemented similarly by multiple unit output layers.

The network realizing the HyperBF approximation scheme has the same structure, depicted in Figure 1.4. Beside the structural equivalence, there are several differences between RBF and HyperBF networks' parameters and learning behaviour, but due the lack of space we refer the Reader to the literature [7,19].

In the Appendix an illustrative example is presented for function approximation with an RBF network.

1.5 The connection of fuzzy algorithms and RBF type schemes

1.5.1 Approximation of fuzzy-to-fuzzy mappings

Comparing the formulae of various fuzzy interpolation techniques and RBF based feedforward networks, some similarities can be detected. Fuzzy rea-

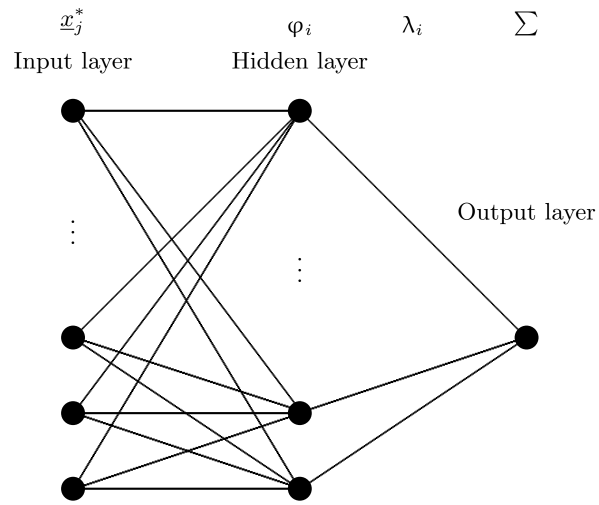

Fig. 1.4. Three-layer feedforward neural network with RBF activation functions in the hidden layer

soning and control systems are approximators on two different levels. They approximate primarily fuzzy-to-fuzzy mappings in the following sense:

$$S : \widetilde{P}(X) \to \widetilde{P}(Y),$$

S is the generalization of the notion of a set-to-set function. As every fuzzy set is described by a membership function, for every given $\alpha \in (0,1]$ and fixed C ("lower" or "upper" end), S realizes a function

$$S_{\alpha,C} : X \to Y,$$

and fuzzy interpolation means the solution of an ordinary interpolation problem for every $S_{\alpha,C}$. The r rules define r vectors $(A_{i\alpha C}, B_{i\alpha C})$, so that the interpolation problem is equivalent to fitting a function to these r control points. Fuzzy interpolation means finding a family of functions $y = F(\underline{x}; \alpha, C)$, $\alpha \in (0,1]$, $C \in \{L, U\}$ interpolating these families of points. For a fixed α-cut and C, the problem to solve is identical with that of ordinary interpolation.

1.5.2 Mamdani like controllers and RBF schemes

On the other hand, if a fuzzy system is considered with crisp observations and defuzzified outputs, the whole system is equivalent to a single approximator. Because the different defuzzification techniques are rather complicated to express in terms of the approximation error involved, in this context we refer to the explicit formulae of the most frequently used fuzzy controllers [13], but will not discuss this aspect here in detail.

Getting back to the first type of problem, it can be recognized that Mamdani control has a similarity to RBF approximation, but is less specific concerning the type of functions used.

Obviously, the original Mamdani method is rather different from the RBF approach, as in expression (1.1) instead of addition a max operator combines the functions height($A^*(\underline{x}), A_j(\underline{x})$), and also the "weighting" is done by min rather than multiplication. So, the formula (1.1) can be written as:

$$F(\underline{x};\alpha,C) = \max_{j=1}^{r}\{\min\{\lambda_j(\alpha,C), A_j(\underline{x})\}\}, \tag{1.13}$$

where $\lambda_j(\alpha,C) = B_{j\alpha C}(y)$ if B^* is calculated point by point. There is a striking analogy with the formulae of the RBF-s, especially as in the theory of possibility measures, the max operator replaces addition and min might play a similar role as multiplication in probability theory. So, the above equation can be interpreted as a "*possibilistic linear combination*" of the consequent function points. An additional essential difference is however the lack of radial behavior.

According to many results, Mamdani type approximators are universal [4,14,24], at least in the sense that their full versions including defuzzification can arbitrarily well approximate an arbitrary continuous function over a compact domain. It is well worth investigating in the future whether this statement remains true for arbitrary continuous fuzzy-to-fuzzy mappings as well.

If Larsen interpolation is applied with centre of gravity defuzzification, the equivalent conclusion can be written point by point as

$$F(\underline{x};\alpha,C) = \sum_{j=1}^{r}\lambda_j(\alpha,C)A_j(\underline{x}), \tag{1.14}$$

where $\lambda_j(\alpha,C)$ is the same as in (1.13) for the Mamdani formula. Here, the only difference to the RBF case is the non-radial nature of the general antecedent function.

Proposition 1. *If in the Larsen algorithm all antecedent functions are symmetric around the centre of the kernel:* $A_j(\underline{\xi}_j - \delta) = A_j(\underline{\xi}_j + \delta)$, *and further, if* $A_j(\underline{x})$ $(j = 1,\ldots,r)$ *differ from each other only by a constant factor, this special fuzzy controller implements an RBF approximator for every fixed* α.

It has been shown in [13], that the transfer function of symmetric Sugeno controllers is identical with the transfer function of Larsen algorithm using the centre of gravity method. Exploiting this functional equivalence, we have the following:

Corollary 1. *Sugeno controllers with symmetric shaped antecedent sets around the centre of the kernel:* $A_j(\underline{\xi}_j - \delta) = A_j(\underline{\xi}_j + \delta)$*, and further, if* $A_j(\underline{x})$ $(j = 1, \ldots, r)$ *differ from each other only by a constant factor, also this special fuzzy controller implements an RBF approximator for every fixed* α*.*

We remark here that Jang achieved similar results for Sugeno controllers in 1993 [10], but he assumed several unnecessary restrictions, e.g. he fixed the shape of membership functions and radial basis functions to be Gaussian of the form:

$$A_j(\underline{x}) = \exp\left(-\frac{\|\underline{x} - \underline{\xi}_j\|^2}{\sigma_j^2}\right) \quad (i = 1, \ldots, r) \tag{1.15}$$

where $\|\cdot\|$ is the Euclidean norm, and σ_j is the common variance belonging to each rules. Proposition 1 does not restrict the shape of the membership functions, just their symmetry are required, hence it also covers the most popular symmetric trapezoidal and triangular membership function shapes.

Beside being symmetric, the antecedents satisfying Proposition 1 have identical shape (called *base shape*) apart from a constant factor. The antecedents belonging to a specific rule have an identical shape which is determined by the stretching (or shrinking) factor of λ_j. In other words, the only degree of freedom, once a base shape of the antecedents is chosen, is the stretching (or shrinking) factor of λ_j in (1.14), which can be different for each n-dimensional antecedent. Figure 1.5 depicts a rule base scheme of triangular membership functions.

Considering the results of [4], it would be interesting to investigate whether in general, any convex and normal fuzzy (CNF) membership function could generate a suitable RBF with positive definite matrix $\underline{\underline{\Phi}}$, where the only necessary condition for the radial basis function is that $\varphi(0) = 1$, $\varphi(z) \to 0$, when $z \to 0$ and is monotone decreasing.

1.5.3 Fuzzy KH interpolators and RBF schemes

Considering now the formulae of the linear KH interpolation,

$$F(\underline{x}; \alpha, C) = \sum_{j=1}^{2} \lambda_j(\alpha, C)\varphi(\underline{x}),$$

where $\varphi(x) = \dfrac{1}{d_{\alpha,C}(A^*(\underline{x}), A_j(\underline{x}))}$ and $\lambda_j(\alpha, C) = \dfrac{B_{j\alpha C}}{\sum_{j=1}^{2} \frac{1}{d_{\alpha,C}(A^*(\underline{x}), A_j(\underline{x}))}}$, is obviously an RBF approach for every fixed level and end point.

Similar constructions apply for the general KH interpolation, where

$$F(\underline{x}; \alpha, C) = \sum_{j=1}^{r} \lambda_j(\alpha, C)\varphi(\underline{x}),$$

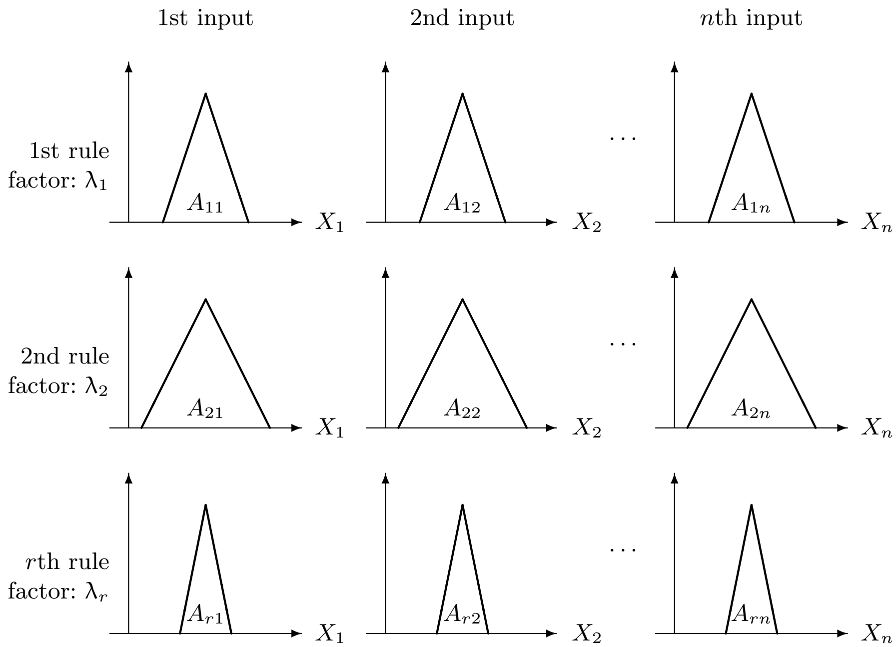

Fig. 1.5. Antecedents of an RBF-equivalent rule base with triangular membership functions

$\varphi(x) = \dfrac{1}{d_{\alpha,C}(A^*(\underline{x}), A_j(\underline{x}))}$ and $\lambda_j(\alpha, C) = \dfrac{B_{j\alpha C}}{\sum_{j=1}^{r} \frac{1}{d_{\alpha,C}(A^*(\underline{x}), A_j(\underline{x}))}}$, for every fixed level and end point.

Analogously, for the stabilized KH interpolator similar constructions also apply, where we take the sth power of the distance function in the previous expressions, where $s \geq n$ and n stands for the number of variables.

Hence we have the following result:

Proposition 2. *The general KH interpolation implements an RBF approximator for every fixed* α. *Here, due to the construction of* $\varphi(\underline{x})$ *and* $\lambda_j(\alpha, C)$, *the vectors* $\underline{\xi}_i$ $(i = 1, \ldots, r)$ *need not be distinct to assure the regularity of the coefficient matrix* $\underline{\underline{\Phi}}$.

A very similar statement can be said about the stabilized version of the KH interpolation, where however the distance function is raised to the power s ($s \geq n$ and n is the number of variables). Consequently we have:

Proposition 3. *The stabilized general KH interpolation also implements an RBF approximator for every fixed* α. *Here, due to the construction of and* $\varphi(\underline{x})$ *and* $\lambda_j(\alpha, C)$, *the vectors* $\underline{\xi}_i$ $(i = 1, \ldots, r)$ *need not be distinct to assure the regularity of the coefficient matrix* $\underline{\underline{\Phi}}$.

1.5.4 Generalization possibilities with HyperBF networks

Let us now investigate how the use of HyperBF schemes can be enhance the equivalent fuzzy controller type, i.e. how it generalize the design parameters of fuzzy controllers.

Let us examine which condition of Proposition 1 can be relaxed with the use of a more general weighted norm. Clearly, the condition on the symmetry of the antecedent fuzzy sets can not be eliminated, because $\underline{x} - \underline{\xi}_j$, the argument of the weighted norm (1.11) is symmetric itself.

If we use weighted norm instead of the regular Euclidean one, the antecedents will have two degrees of freedom. The first one is the same as described in Figure 1.5. The second one comes from the norm weighting matrix, $\underline{\underline{C}}$ in (1.11). As we already mentioned, with diagonal $\underline{\underline{C}}$ we can give different weights to each variable. Let assume that we are given Gaussian membership functions of form (1.15) and let

$$\underline{\underline{C}} = \operatorname{diag}\left(\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_n}\right).$$

Then

$$A_{j1}(x_1) = \exp\left(-\frac{\|x_1 - \xi_{j1}\|}{\sigma_1^2}\right) \qquad \ldots$$
$$A_{jn}(x_n) = \exp\left(-\frac{\|x_n - \xi_{jn}\|}{\sigma_n^2}\right)$$

that is a different variance can be assigned to each dimension.

More complicated affine transformations of the input space can be achieved with more sophisticated norm weighting matrices (cf. [7], p. 282).

This means that the antecedents should be symmetric, in each dimension there is a stretching/shrinking factor and for each rule there is another stretching/shrinking factor.

Finally, we can state that with the HyperBF approximation scheme, the generalization of the RBF approximation scheme, we can implement slightly more general Larsen and Sugeno type fuzzy controllers. This is formalized as:

Proposition 4. *If in the Larsen or the Sugeno algorithm all antecedent functions are symmetric around the centre of the kernel:* $A_j(\underline{\xi}_j - \delta) = A_j(\underline{\xi}_j + \delta)$, *and further, if projections* $A_{ji}(\underline{x}_i)$ *of* $A_j(\underline{x})$ $(j = 1, \ldots, r;\ i = 1, \ldots, n)$ *differ from each other only in a constant factor, and* $A_j(\underline{x})$ $(j = 1, \ldots, r)$ *also differ from each other by a constant factor, this special fuzzy controller implements an HyperBF approximator for every fixed* α.

1.5.5 On the advantages of neuro-fuzzy techniques

According to these result, these fuzzy controllers can be implemented as RBF or HyperBF networks. With a neuro-fuzzy method, we can exploit the advantageous properties of both neural and fuzzy techniques. The two most popular neuro-fuzzy techniques are the Jang's ANFIS (Adaptive Network-based Fuzzy Inference System) [9], and Nauck and Kruse's works (NEFCON, NEFCLASS and the extended NEFPROX models) [17]. The former implements a Takagi–Sugeno controller with an adaptive feedforward neural network, and uses a hybrid learning algorithm (gradient method for the input and least square error method for the output parameters) for optimization; the latter implements a Mamdani-like method using a neuro-fuzzy architecture with supervised learning to determine the structure and the parameters of a fuzzy system. NEFCON and NEFCLASS are for control and classification purposes, while NEFPROX is more general and can be used for any application based on function approximation.

In the case where one has certain knowledge about the modelled system, e.g. interpreted as fuzzy rules, this information can be used to determine the initial parameters of a neural network. By means of the training data and the learning algorithm applied to the network, the corresponding fuzzy rule base can be tuned, refining the model of the system. Such an approach results in a well-tuned model of the system (naturally to some extent depending on the reliability and appropriateness of the training data), while the tuned rule base model of the system can still provide a linguistic description of the system, thus it assures the tracing of the system by a human operator. The linguistic description of the system gives a possible tool for quality control in certain cases: if the training data set is noisy or degenerate, a human operator can discover these kinds of mistakes much easier based on a linguistic description than merely on the transfer function calculated by the network.

In the case when only input-output sample data are available, a combined neuro-fuzzy approach helps in the extraction of fuzzy rules from the data. Once the rules are extracted from the training data set the rule based description of the system is available.

The advantage of the use of a fuzzy rule interpolation approach compared to the classical fuzzy reasoning approach is that one can have such input-output sample data sets which does not cover the whole range of the input space, e.g. in large regions of the input space the appropriate output remains unknown. As a consequence, the extracted rule base can contain gaps (a sparse rule base). In such a case the fuzzy controllers operating on dense rule bases can not be applied. As an industrial example we refer to well-log analysis in petroleum industry (see [25]).

1.6 Conclusion

In this paper new examples have been given for the approximate functional equivalence of fuzzy and neural models. We pointed out that under certain conditions Larsen type controllers and Sugeno type controllers implement RBF approximation schemes.

We have also shown that the (stabilized) general fuzzy KH interpolation and the RBF approximation scheme (usually implemented by RBF networks), both being considered as universal approximators, can be connected straightforwardly. The general fuzzy KH interpolators can be considered as a special RBF approximation scheme for fixed α-cuts $\alpha \in (0,1]$, and $C \in \{L, U\}$.

We also pointed out to what extent the use of the weighted norm can generalize the implementation of Larsen controllers. Finally, some of the possible advantages of the combination of fuzzy and neural network techniques were outlined.

Appendix: An illustrative example for approximation

Although it is impossible to show experimentally (through numerous examples) that a given method is a universal approximator, here we present a simple example to illustrate how the approximation works with the aforementioned soft computing techniques.

The function we picked for approximation is $\sin(7x)$ on the unit interval. The function is given at 51 equidistant located sample points: $\{x_i = 0.02i | i \in [0, 50]\}$. We compared RBF network and Takagi–Sugeno model with exponentional (bell-shaped) membership functions. We used the mean squared error for evaluating the performance of the analyzed methods. We set the required accuracy to 0.01 in both cases.

Fig. 1.6. Approximation with RBF network

Figure 1.6 depicts the sample points and approximating function. The network has one input, six hidden, and one output neuron. The approximation was performed with MATLAB neural network toolbox functions. The performance index is $2.48 \cdot 10^{-5}$. The elapsed time was about 1.5 sec.

We accomplished the Takagi–Sugeno fuzzy approximation with several choices for the number of membership functions. We used Robert Babuška's [1] FM (Fuzzy Modeling) structure under MATLAB to perform the approximation.

Table 1.1 summarizes the results of these trials. Figure 1.7 shows the approximation performed with Takagi–Sugeno fuzzy model and 5 membership function. The membership functions themselves, and the corresponding rule outputs (called local models on the upper picture) are depicted in figure 1.8.

Fig. 1.7. Approximation with TS fuzzy model using 5 membership functions

Fig. 1.8. Membership functions and local models with TS approximation

Table 1.1. Characteristics of approximation with different number of membership functions

Number of membership function	Performance index	Elapsed time (s)
2	0.04	1.3
3	0.0058	1.7
5	$5.93 \cdot 10^{-4}$	≈ 2
10	$2.87 \cdot 10^{-5}$	≈ 5
15	$3.43 \cdot 10^{-6}$	≈ 5.6
51	$2.89 \cdot 10^{-31}$	≈ 14

One can observe that the Takagi–Sugeno modeling provides worse result in terms of the accuracy and the required time as well. This is mainly due to the underlying clustering method used in the FM for identifying fuzzy membership functions. On the other hand, this method gives us an insight to understand how the approximation works and also allows us to introduce a linguistic representation of the problem. We remark that with slightly more time spent on the calculation (about 5 s) we obtain a good approximation benchmark. If we set the number of membership functions equal to the number of sample data points we can get errors as small as 10^{-31}. (This value might not be reliable as the computation contains some badly conditioned matrices).

Acknowledgement

The authors want to thank to János Levendovszky for his valuable suggestions on the connection of KH interpolation and RBF functions, and to János Abonyi for providing us the Fuzzy Model program codes.

References

1. Robert Babuška. Fuzzy modeling for matlab. http://lcewww.et.tudelft.nl/ cr-web/software/index.html.
2. P. Baranyi and L. T. Kóczy. A general and specialized solid cutting method for fuzzy rule interpolation. *BUSEFAL*, 67:13–22, 1996.
3. D. S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.
4. J. L. Castro. Fuzzy logic controllers are universal approximators. *IEEE Trans. on System, Man, and Cybernetics*, 25(4):629–635, 1995.
5. A. C. Faul and M. J. D. Powell. Krylov subspace methods for radial basis function interpolation. Technical Report DAMPT 1999/NA11, University of Cambridge, 1999. 25p.
6. T. D. Gedeon and L. T. Kóczy. Conservation of fuzziness in rule interpolation. In *Proc. of the Symp. on New Trends in Control of Large Scale Systems*, volume 1, pages 13–19, Herľany, 1996.
7. S. Haykin. *Neural Networks: A Comprehensive Foundation.* Prentice Hall, Upper Saddle River, New Jersey, 2nd edition, 1999.
8. K. Hlavačková and R. Neruda. Radial basis functions. Technical report, Institute of Information and Computer Science, Czech Academy of Science, Prague, w.y.
9. J-S. R. Jang. ANFIS: adaptive-network based fuzzy inference system. *IEEE Trans. on System, Man, and Cybernetics*, 23(3):665–685, 1993.
10. J-S. R. Jang. Functional equivalence of radial basis function networks and fuzzy inference systems. *IEEE Trans. on Neural Networks*, 4(1):156–159, 1993.
11. L. T. Kóczy. Algorithmic aspects of fuzzy control. *Internat. J. of Approx. Reason.*, 12:159–217, 1995.
12. L. T. Kóczy and K. Hirota. Approximate reasoning by linear rule interpolation and general approximation. *Internat. J. Approx. Reason.*, 9:197–225, 1993.
13. L. T. Kóczy and M. Sugeno. Explicit functions of fuzzy control systems. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 4:515–535, 1996.
14. B. Kosko. Fuzzy systems as universal approximators. In *Proc. of the 1st IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'92)*, pages 1153–1162, San Diego, 1992.
15. E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. of Man Machine Studies*, 7(1):1–13, 1975.
16. C. A. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Construct. Approx.*, 2:11–22, 1986.
17. D. Nauck and R. Kruse. Neuro-fuzzy systems for function approximation. *Fuzzy Sets and Systems*, 101:261–271, 1999.
18. J. Park and I. W. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation*, 3:246–257, 1991.
19. T. Poggio and F. Girosi. Networks and the best approximation property. *Biological Cybernetics*, 63:169–176, 1990.
20. Y. Shi and M. Mizumoto. Some considerations on Kóczy's interpolative reasoning method. In *Proc. of the 4th IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE/IFES'95)*, pages 2117–2122, Yokohama, 1995.

21. T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. on System, Man, and Cybernetics*, 15(1):116–132, 1985.
22. D. Tikk and P. Baranyi. Comprehensive analysis of a new interpolation method. To appear in *IEEE Trans. on Fuzzy Systems*.
23. D. Tikk, I. Joó, L. T. Kóczy, P. Várlaki, B. Moser, and T. D. Gedeon. Stability of interpolative fuzzy KH-controllers. To appear in *Fuzzy Sets and Systems*.
24. L. X. Wang. Fuzzy systems are universal approximators. In *Proc. of the 1st IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'92)*, pages 1163–1169, San Diego, 1992.
25. K. W. Wong and T. D. Gedeon. Petrophysical properties prediction using self-generating fuzzy rules inference system with modified α-cut based fuzzy interpolation. Submitted to *ICONIP 2000* to be held in Taejon, Korea, 2000.
26. L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.
27. L. A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. on System, Man, and Cybernetics*, 1(1):28–44, 1973.