On Fuzzy Controllers Having Radial Basis Transfer Functions*

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

[1] Department of Telecommunication & 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, 6148 W.A., Australia

Abstract. This works establish a link between certain fuzzy controllers and radial basis function networks. We deal with the two soft computing techniques and compare them from the approximation capability point of view. The fact that the approximation behaviour of these systems are similar establishes the ground of analysing their possible functional equivalence. We show that certain fuzzy systems can be considered as radial basis function (RBF) approximation scheme, having the same transfer function. This allows to implement those fuzzy systems by RBF networks, and hence, establishing a learning algorithm for them and improving their approximation capabilities. We also points out that the classes of the corresponding fuzzy system can be generalized to some extent if we use general basis function with weighted norm.

1 Introduction

The approximation capabilities of soft computing techniques has been investigated intensively in the last 10–15 years. The approximation behaviour of these techniques was found to be similar: most of the general fuzzy systems and neural networks are universal approximators on a compact subset of continuous functions with respect to the supremum or L_p norm $p \in [1, \infty)$.

These results reveal that the set of transfer functions which can be approximated by fuzzy systems or neural networks more or less the same. However, the way they captures the underlying transfer functions 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. Neuro-fuzzy systems, which link these techniques, enable us to combine their advantageous properties, and hence, we can obtain better behaving tools. This can be backed from the practical side by the numerous successful applications (see e.g. [1–3]).

* 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.

L. Reznik et al. (eds.), *Soft Computing in Measurement and Information Acquisition*

In this paper we will show that several fuzzy controllers implement one of the typical neural networks (having radial basis activation functions). One type of those fuzzy systems is the KH interpolative controller to be recalled in Section 2. We will briefly summarize the mathematical background of (RBF) network in Section 3. We present how the implementing fuzzy systems vary if we use weighted norm in the network.

2 Interpolation in Sparse Fuzzy Rule Bases

The classical approaches of fuzzy control deal with dense rule bases where the universe of discourse is α-covered (at any point x from the input set, there exists a fuzzy set A, for which $A(x) \geq \alpha$ holds, usually $\alpha \geq 0.5$) by the antecedent fuzzy sets of the rule base in each dimension. 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 classical fuzzy reasoning methods (Zadeh's CRI [4], Mamdani [5], Larsen [6], and Takagi–Sugeno [7]) 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

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

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), the fuzzy similarity set can be defined as

$$\begin{aligned}\widetilde{s}(A_1, A_2) &= \{s_{\alpha,C}(A_1, A_2),\ \alpha \in (0,1], C \in \{L, U\}\} \\ &= \left\{\frac{1}{d_{\alpha,C}(A_1, A_2)}, \alpha \in (0,1]\right\}\end{aligned} \tag{2}$$

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):

$$D(A^*, A_1) : D(A^*, A_2) = D(B^*, B_1) : D(B^*, B_2). \tag{3}$$

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 $B_1 \prec B_2$. 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 (3) 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 (2) 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$(A_j \wedge A^*)$ (height$(A) = \max_x A(x)$ is the height of a fuzzy set; $\wedge$ denotes an appropriate t-norm) 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 [8]), which also functions in sparse rule bases. Hence the basic formula of the Mamdani type algorithm,

$$B^* = \max_j \{\min\{\text{height}(A^*(\underline{x}), A_j(\underline{x})), B_j(y)\}\}, \tag{4}$$

is replaced by

$$B^*_\alpha = \frac{\sum_{j=1}^{r} s_\alpha(A^*, A_j) B_{j\alpha C}}{\sum_{j=1}^{r} s_\alpha(A^*, A_j)} = \frac{\sum_{j=1}^{r} \frac{1}{d_{\alpha,C}(A^*,A_j)} B_{j\alpha C}}{\sum_{j=1}^{r} \frac{1}{d_{\alpha,C}(A^*,A_j)}} \tag{5}$$

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. [8]). 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 [9] 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 [10].) In this sense, fuzzy rule based controllers can be considered as multivariate real function approximators.

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 [11]. 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

$$s(A^*, A_j) = \frac{1}{\int\limits_{\alpha_L \in (0,1] \wedge \alpha_u \in (0,1]} d_{\alpha,C}(A^*, A_j) \mathrm{d}\alpha}. \tag{6}$$

In the case of trapezoidal membership functions (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

$$s_{\alpha,C}(A^*, A_j) = \frac{\frac{1}{d^n_{\alpha,C}(A^*,A_j)}}{\sum_{j=1}^{r} \frac{1}{d^n_{\alpha,C}(A^*,A_j)}}, \tag{7}$$

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 [12]. Due to this result, the input-output function of the general KH interpolation obtained by means of the radical similarity defined in (7),

$$B^*_{\alpha,C} = \frac{\sum_{j=1}^{r} \frac{1}{d^n_{\alpha,C}(A^*,A_j)} B_{j\alpha C}}{\sum_{j=1}^{r} \frac{1}{d^n_{\alpha,C}(A^*,A_j)}} \tag{8}$$

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 (see e.g. [13–15]). 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 sort of 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 [12]. 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 being able to approximate fuzzy-to-fuzzy mappings on the Cartesian product of compact universes of discourse.

There are further methods for fuzzy rule interpolation that we do not discuss here in detail (cf. [16–18]).

3 Radial Basis Functions in Neural Networks

3.1 Mathematical background

An alternative approach in soft computing field is the artificial neural network model. Three-layer feedforward neural networks using *radial basis functions* are similarly function approximators as are rule based fuzzy systems. Park and Sandberg proved [19] that any continuous function f could be arbitrarily well approximated on a compact subset $X \subset \mathbb{R}^n$ in L_∞ norm:

$$f(\underline{x}) = \sum_{i=1}^{r} \lambda_i \varphi(\|\underline{x} - \underline{\xi}_i\|) \tag{9}$$

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, $\{y_i = F(\underline{x}_i),\ i = 1, \dots, r\}$, by substituting them into (9), a set of linear equations for is obtained [20]:

$$\begin{pmatrix} y_1 \\ \vdots \\ y_r \end{pmatrix} = \begin{pmatrix} \varphi_{11} & \dots & \varphi_{1r} \\ \vdots & \ddots & \vdots \\ \varphi_{r1} & \dots & \varphi_{rr} \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_r \end{pmatrix} \tag{10}$$

where

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

By denoting $\underline{\underline{\Phi}} = \begin{pmatrix} \varphi_{11} & \cdots & \varphi_{1r} \\ \vdots & \ddots & \vdots \\ \varphi_{r1} & \cdots & \varphi_{rr} \end{pmatrix}$, $\underline{\lambda} = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_r \end{pmatrix}$, and $\underline{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_r \end{pmatrix}$, (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. [21]), if $\underline{\xi}_i$ $(i = 1, \ldots, r)$ 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 one positive one [22].

The RBF approach can be extended to multiple output mappings, and more general approximation, where the number of known points rather larger than the number of functions φ_i, using the Moore–Penrose pseudo inverse (cf. [20,23]):

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

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

3.2 HyperBF approximation with weighted norm

The usually Euclidean norm function used in the RBF approximation (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 [24]:

$$\|\underline{x}\|_C^2 = (\underline{\underline{C}}\underline{x})^T(\underline{\underline{C}}\underline{x}) = \underline{x}^T\underline{\underline{C}}^T\underline{\underline{C}}\underline{x} \tag{11}$$

where $\underline{\underline{C}}$ is the *norm weighting matrix* with size $n \times n$, n being the dimension of $\underline{x}$.

We can generalize the RBF approximation scheme by using the weighted norm defined in (11) as

$$f(\underline{x}) = \sum_{i=1}^{r} \lambda_i \varphi(\|\underline{x} - \underline{\xi}_i\|_C) \tag{12}$$

which scheme are called *generalized radial*, or *hyper basis function* approximation scheme [24].

The use of the weighted norm can be interpreted as a simple *affine transformation* of the input space. As a special case, the approximation (12) contains the original RBF scheme (9), when $\underline{\underline{C}} = I_n$, the n dimensional identity matrix. We will return to HyperBF approximation scheme in the next section.

3.3 The RBF network

Three-layer feedforward neural networks using radial basis functions and with fully interconnected layers are similarly suitable function approximators as the fuzzy systems. 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 $i \in \{1, \dots, r\}$ indicates the subscript of the corresponding unit centre, and $j \in \{1, \dots, n\}$ 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). Multiple unit output layers can similarly implement multiple output variable cases.

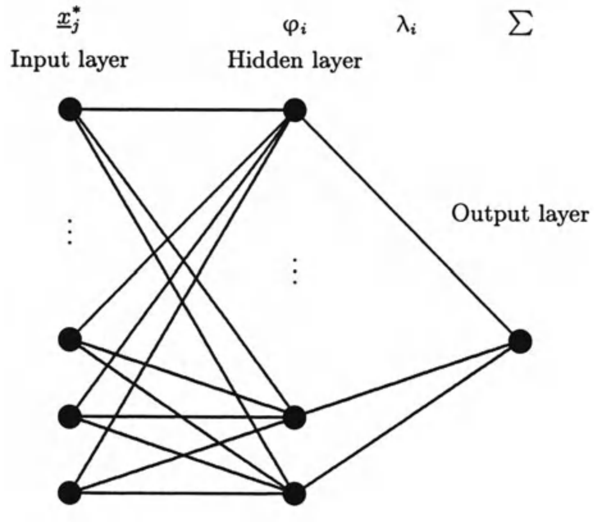

Fig. 1. Three-layer feedforward neural network with RBF activation functions in the hidden layer

4 Fuzzy Controllers Implementing RBF Networks

4.1 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 reasoning 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.

4.2 Classical 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 of the different defuzzification techniques are rather complicated to be expressed in terms of the approximation error involved, in this context we refer to the explicit formulae of the most frequently used fuzzy controllers [10], 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 not addition but a max operator combines the rule premises and the observation (i.e. functions height$(A^*(\underline{x}), A_j(\underline{x}))$, and also the "weighting" is done by min rather than multiplication. So, the formula (4) can be written like:

$$F(\underline{x}; \alpha, C) = \max_{j=1}^{r}\{\min\{\lambda_j(\alpha, C), A_j(\underline{x})\}\}, \tag{13}$$

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 the 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 [25–27], 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

$$F(\underline{x};\alpha,C) = \sum_{j=1}^{r} \lambda_j(\alpha,C) A_j(\underline{x}), \tag{14}$$

where $\lambda_j(\alpha, C)$ is the same as in (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:* $A_i(\underline{\xi}_i - \delta) = A_i(\underline{\xi}_i + \delta)$, *and further, if* $A_i(\underline{x})$ $(i = 1, \ldots, r)$ *differ from each other only in a constant factor, this special fuzzy controller implements an RBF approximator for every fixed* α.

It has been shown in [10], that the transfer function of symmetric Sugeno controllers [11] is identical with the transfer function of Larsen algorithm using the centre of gravity method. Exploiting this functional equivalence, we can have the following

Corollary 1. *Sugeno controllers with symmetric antecedent sets with respect to the centre of the kernel:* $A_i(\underline{\xi}_i - \delta) = A_i(\underline{\xi}_i + \delta)$, *and further, if* $A_i(\underline{x})$ $(i = 1, \ldots, r)$ *differ from each other only in a constant factor, this special fuzzy controller implements an RBF approximator for every fixed* α, *as well.*

Considering the results of [27], 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. We conjecture that the answer of this question is rather negative.

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

Similar constructions apply for the general KH interpolation, where

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

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

Analogously, for the stabilized KH interpolator also similar constructions 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 to 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 to be distinct to assure the regularity of the coefficient matrix $\underline{\underline{\Phi}}$.*

Observe that KH interpolators implement RBF network straightforwardly: no restrictions apply on the shape or location of the involved membership functions.

4.4 Generalization possibilities with HyperBF networks

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

Let us examine which condition of Proposition 1 can be relaxed with the use of more general weighted norm. Clearly, the condition on the symmetry of the antecedent fuzzy sets can not be omitted, because the weighted norm (11) is symmetric, as well. The antecedents satisfying Proposition 1 are symmetric, and in each dimension, their projections has a fixed shape, termed as *base shape*, from which the antecedents can differ only in a constant factor. In

Proposition 1 the only degree of freedom, once a base shape of the antecedents is chosen, is the stretching (or shrinking) factor of λ_j in (14), which can be different for each n-dimensional antecedents. This only allows the projections of an arbitrary antecedent A_j to be different from the base shape of the dimension with the fixed stretching/shrinking factor defined by the coefficient λ_j.

If we use weighted norms instead of the regular Euclidean one, the antecedents have two degrees of freedom. One which enables them to be different to a constant extent for each n-dimensional antecedents (λ_j), and the other which enables them to be different to a constant extent in each dimension (determined by the weighting matrix C). It means that the antecedents should be symmetric, in each dimension there is a base shape of the antecedents' projection, but from this shape the projections can differ with a constant factor for each antecedents, and in each input dimension a further stretching/srinking factor can be given which enables the input spaces to be different by this scaling factor. The scaling factors of the input spaces are determined with the weighting matrix C (see 11). 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 summarized in the next

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

4.5 On the advantages of neuro-fuzzy techniques

According to these result, general Larsen and KH-controllers can be implemented by RBF or HyperBF networks. With a neuro-fuzzy method, we can exploit the advantageous properties of both neural and fuzzy techniques as described in the next paragraphs.

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, this way, 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 degenerated, a human operator can discover this kind 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.

We emphasize that the advantage of using fuzzy rule interpolation approaches compared to the classical fuzzy reasoning approach is not solely that the former is capable to handle sparse rule base, but — as it has been shown in the previous sections — also less restriction applies concerning the implementation by RBF networks. We remark that sparse rule bases may occur extensively when automatic rule extraction and generation is used. In such cases when the input-output sample data set does not cover the whole range of the input space, the appropriate output value may remain unknown for large uncovered input regions. As a consequence, the extracted rule base is sparse, so fuzzy controllers operating on dense rule bases can not be applied. A practical example which uses fuzzy interpolation (different from the KH method) to solve this problem (applied for well-log analysis in petroleum industry) is presented in [28].

5 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 scheme.

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 implementing Larsen controllers. Finally, some of the possible advantages of the combination of fuzzy and neural techniques were outlined.

Acknowledgement

The authors want to thank to János Levendovszky for his valuable suggestions on the connection of KH interpolation and RBF functions.

References

1. J.-S. R. Jang, C.-T. Sun and E. Mizutani. (1997) Neuro Fuzzy and Soft Computing. Prentice Hall, Upper Saddle River, NJ.
2. P. M. Wong, K. W. Wong, C. C. Fung, and T. D. Gedeon. (1997) A neural-fuzzy technique for interpolating spatial data via the use of learning curve. Biological and Artificial Computation: From Neuroscience to Technology (eds: J. Mira, R. Moreno-Diaz, J. Cabestany), Springer-Verlag, 323–329.
3. K. W. Wong, D. Myers and C. C. Fung. (1999) A generalised neural-fuzzy well log interpretation model with a reduced rule base. The Sixth International Conference of Neural Information Processing ICONIP, Perth, Australia, 188-191.
4. L. A. Zadeh. (1973) Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans. on SMC **3**, 28–44.
5. E. H. Mamdani and S. Assilian. (1975) An experiment in linguistic synthesis with a fuzzy logic controller. Int. J. of Man Machine Studies **7**, 1–13.
6. P. M. Larsen. (1980) Industrial application of fuzzy logic control. Int. J. of Man Machine Studies **12**(4), 3–10.
7. M. Sugeno and T. Takagi. (1985) Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. on SMC **15**, 116–132.
8. L. T. Kóczy and K. Hirota. (1993) Approximate reasoning by linear rule interpolation and general approximation. Internat. J. Approx. Reasoning **9**, 197–225.
9. Y. Shi and M. Mizumoto. (1995) Some considerations on Kóczy's interpolative reasoning method. Proc. of the FUZZ-IEEE/IFES'95, Yokohama, 2117–2122.
10. L. T. Kóczy and M. Sugeno. (1996) Explicit functions of fuzzy control systems. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **4**, 515–535.
11. M. Sugeno. (1985) An introductory survey of fuzzy control. Information Science **36**(1–2), 59–83.
12. 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.
13. D. Shepard. (1968) A two dimensional interpolation function for irregularly spaced data. Proc. of the 23rd ACM International Conference, 517–524.
14. J. Szabados. (1991) Direct and converse approximation theorems for Shepard-operator. J. Approx. Theory and its Appl. **7**, 63–76.
15. B. Della Vecchia and G. Mastroianni and J. Szabados. (1996) Balázs–Shepard operators on infinite intervals. Annales Uni. Sci. Budapest, Sectio Comput. **16**, 93–102.
16. T. D. Gedeon and L. T. Kóczy. (1996) Conservation of fuzziness in rule interpolation. Proc. of the Symp. on New Trends in Control of Large Scale Systems, Herlany, vol. 1, 13–19.
17. P. Baranyi and T. D. Gedeon. (1996) Rule interpolation by spatial geometric representation. Proc. of the IPMU'96, Granada, 483–488.
18. D. Tikk and P. Baranyi. Comprehensive analysis of a new fuzzy rule interpolation method. To appear in IEEE Trans. on Fuzzy Systems.
19. J. Park and I. W. Sandberg. (1991) Universal approximation using radial-basis-function networks. Neural Computation **3**, 246–257.

20. K. Hlavačková and R. Neruda. (without year) Radial basis function networks. Inst. of Information and Computer Science, Czech Academy of Sciences, Prague, report.
21. A. C. Faul and M. J. D. Powell. (1999) Krylov subspace methods for radial basis function interpolation. DAMPT 1999/NA11, University of Cambridge, 25p.
22. C. A. Micchelli. (1986) Interpolation of scattered data: Distance matrices and conditionally positive definite functions. Constructive Approximation **2**, 11–22.
23. D. S. Broomhead and D. Lowe. (1988) Multivariable functional interpolation and adaptive networks. Complex Systems **2**, 321–355.
24. T. Poggio and F. Girosi. (1990) Networks and the best approximation property. Biological Cybernetics **63**, 169–176.
25. L. X. Wang. (1991) Fuzzy systems are universal approximators. Proc. of the IEEE Int. Conf. on Fuzzy Systems, San Diego, 1163–1169.
26. B. Kosko. (1991) Fuzzy systems as universal approximators. Proc. of the IEEE Int. Conf. on Fuzzy Systems, San Diego, 1153–1162.
27. J. Castro. (1995) Fuzzy logic controllers are universal approximators. IEEE Trans. on SMC **25**, 629–635.
28. 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 Intelligent Systems and Application (ISA'2000, Sydney).