

Extending the Functional Equivalence of Radial Basis Function Networks and Fuzzy Inference Systems

Kenneth J. Hunt, Roland Haas, and Roderick Murray-Smith

Abstract—We establish the functional equivalence of a generalized class of Gaussian radial basis function (RBF's) networks and the full Takagi–Sugeno model of fuzzy inference. This generalizes an existing result which applies to the standard Gaussian RBF network and a restricted form of the Takagi–Sugeno fuzzy system. The more general framework allows the removal of some of the restrictive conditions of the previous result.

I. INTRODUCTION

WE consider the functional equivalence of radial basis function (RBF) networks and a class of fuzzy inference systems. The structure of the “standard” RBF network is described in Broomhead and Lowe [1] and Moody and Darken [2]. The class of fuzzy systems we consider is based on the model of Takagi and Sugeno [3], and we will refer to this as the TS-model of fuzzy inference. The conditions under which the standard Gaussian RBF network and a restricted form of the TS-model of fuzzy inference are functionally equivalent were established by Jang and Sun [4].

Our aim in this paper is to generalize the Jang–Sun result by removing some of the restrictions on the class of RBF networks and fuzzy systems to which the result applies. We achieve this aim by defining a generalized class of RBF networks. Briefly, the expanded class of RBF networks admits the use of local models in network output links, has ellipsoidal basis functions, and can directly omit redundant elements of the input vector. We also show how the equivalence result applies to a wider class of fuzzy systems; we discuss how a large class of fuzzy systems can be transformed to the TS-model.

Local model networks were introduced by Jones *et al.* [5] and further developed by Johansen and Foss [6]; their relation to fuzzy systems was discussed in Foss and Johansen [7]. The functional equivalence of the TS-model and spline-based networks has been considered by Hunt *et al.* [8]. The practical relevance of the functional equivalence result is that the learning algorithm of one paradigm can be used to train models expressed in the other paradigm. An important aspect is the ability to use fuzzy *a priori* knowledge to prestructure a network. The equivalence also allows models of one paradigm to be interpreted in the language of the other. Finally, implementation can be done in either form.

Manuscript received February 4, 1994; revised May 11, 1995.

K. J. Hunt is with Daimler-Benz AG, D-10559 Berlin, Germany.

R. Haas and R. Murray-Smith are with Daimler-Benz AG Systems Technology Research, D-10559 Berlin, Germany.

Publisher Item Identifier S 1045-9227(96)01239-8.

II. RBF NETWORKS

In Section II-A we describe the generalized RBF network. The conditions under which it reduces to the standard network are given in Section II-B. The main features of the generalized RBF network are:

- 1) Each processing unit in the network receives as input possibly only a subset of the complete network input vector (see (1) and the example in Section IV-D). This means that irrelevant elements of the input vector can be easily left out of a RBF unit. This is important as not all elements of the input vector need necessarily appear in the premise of the corresponding fuzzy if–then rule.
- 2) The output links of the network consist of local, possibly nonlinear, models which process the network input vector [these are defined by the functions $\theta_i(\vec{x})$ in (1)].¹ The standard RBF network has constant weights connecting to the output unit. This extends the applicability of the standard TS-model of fuzzy inference which effectively has linear local models in the consequent of each fuzzy if–then rule.
- 3) The Gaussian processing units in the network have differing widths in each dimension of the input vector. This results in ellipsoidal basis functions and effectively “decouples” the corresponding basis functions in different dimensions. As a result the shapes of the corresponding membership functions in different dimensions are also decoupled.

A. Generalized RBF Description

The type of network under consideration is described by

$$y = f(\vec{x}) = \sum_{i=1}^{n_\theta} \theta_i(\vec{x}) \phi_i(\vec{x}_i) = \vec{\theta}' \vec{\phi} \quad (1)$$

where $\vec{\cdot}$ denotes a vector and $'$ is the transpose operator. Here, $y \in \mathcal{R}$ is the network output² and $\vec{x} \in \mathcal{R}^{n_x}$ is the input vector. The network has n_θ nonlinear processing units and the nonlinearity of the i th unit is represented by the function $\phi_i(\vec{x}_i)$ with $\vec{x}_i \in \mathcal{R}^{n_{x_i}}$. In general each unit input vector \vec{x}_i is a subset of the network input vector \vec{x} , i.e., $\vec{x}_i \subset \vec{x}$ and $n_{x_i} \leq n_x$.

The output of each processing unit is multiplied by the weighting function $\theta_i(\vec{x})$ and these values are summed to form the network output. In the vector form of (1) we note that

¹For a two-dimensional input vector linear local models could have the form $\theta_i(\vec{x}) = k_{i,1}x_1 + k_{i,2}x_2$ with $k_{i,1}$ and $k_{i,2}$ constants.

²In this paper we consider for simplicity only scalar outputs and note that extension to the multioutput case is straightforward.

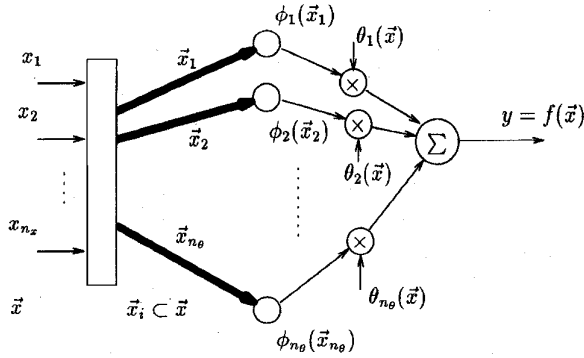


Fig. 1. Generalized basis function network.

$\vec{\theta}, \vec{\phi} \in \mathcal{R}^{n_\theta}$. A normalized form of the network is sometimes used (see [9] for discussion of the effects of normalization) and this is described by

$$y = f(\vec{x}) = \frac{\sum_{i=1}^{n_\theta} \theta_i(\vec{x}) \phi_i(\vec{x}_i)}{\sum_{i=1}^{n_\theta} \phi_i(\vec{x}_i)} \quad (2)$$

The flow of computation in (1) can be represented graphically as the network in Fig. 1 (where the thick lines represent vectors). Such graphical representations of the flow of computation in parameterized interconnections of simple nonlinear functions are popularly referred to as “artificial neural networks” or simply “neural networks.”³ The processing units ϕ_i in (1) are known as basis functions and following tradition we refer to nonlinear parameterization of the form (1) as basis function networks (bland) or, when RBF’s such as the Gaussian form are utilized (see (3) or (7) later), as RBF networks.

The basis function network of (1) represents a generalized form of network first, since the weights $\theta_i(\cdot)$ are functions of the input vector \vec{x} .⁴ We call this type of network a local model network (see [6], [10], and [11] for more on local models). Second, each processing unit acts on only a subset of the input vector. Each of the functions $\theta_i(\vec{x})$ can be viewed as local models whose validity is defined by the activation value $\phi_i(\vec{x}_i)$. Thus the network, through the locally active functions $\phi_i(\vec{x}_i)$, partitions the input space into n_θ operating regions on each of which a local model is defined. The network smoothly joins these local models together through interpolation to form the overall global model $f(\vec{x})$.

A common form of basis function is the radial Gaussian form. A generalized form of Gaussian basis function is described by

$$\phi_i(\vec{x}_i) = \exp[-(\vec{x}_i - \vec{c}_i)' \Delta_i (\vec{x}_i - \vec{c}_i)] \quad (3)$$

Each basis function has two parameters: the center vector $\vec{c}_i \in \mathcal{R}^{n_{x_i}}$ and a width matrix $\Delta_i \in \mathcal{R}^{n_{x_i} \times n_{x_i}}$. The width

³This comes about due to the loose analogy with biological neural networks.

⁴In standard basis function networks the weights are scalar, i.e., $\theta_i(\vec{x}) = \theta_i$ —see Section II-B in the sequel.

parameter is a diagonal matrix of the form

$$\Delta_i = \begin{bmatrix} \frac{1}{\sigma_{i1}^2} & 0 & \dots \\ 0 & \frac{1}{\sigma_{i2}^2} & \dots \\ \vdots & \dots & \ddots \\ & & & \frac{1}{\sigma_{in_{x_i}}^2} \end{bmatrix} \quad (4)$$

The activation of each unit $\phi_i(\vec{x}_i)$ depends on the radial distance of the unit input vector \vec{x}_i from the center of the unit \vec{c}_i . Each dimension j of the distance vector $(\vec{x}_i - \vec{c}_i) \in \mathcal{R}^{n_{x_i}}$ is further weighted by a width parameter $\frac{1}{\sigma_{ij}^2}$, with $j = 1 \dots n_{x_i}$. The basis function described above is generalized in the sense that standard Gaussian basis functions apply the same width parameter in each dimension and each unit processes the whole input vector (see Section II-B below). Due to the definitions (3)–(4) the basis functions are defined on hyperellipsoids in the input space (as a result of the different width in each direction) as opposed to hyperspheres as in standard Gaussian RBF’s.

B. Standard RBF Network

We now state the conditions under which the standard Gaussian RBF network is recovered from the generalized form defined above. There are three conditions under which this is achieved:

- 1) Each basis function processes the whole network input vector, i.e., $n_{x_i} = n_x$ and $\phi_i(\vec{x}_i) = \phi_i(\vec{x})$, for $i = 1 \dots n_\theta$.
- 2) The local models $\theta_i(\vec{x})$ are constants, i.e., $\theta_i(\vec{x}) = \theta_i$. In this case the basis function network (1) or (2) becomes

$$y = f(\vec{x}) = \sum_{i=1}^{n_\theta} \theta_i \phi_i(\vec{x}) \quad (5)$$

or

$$y = f(\vec{x}) = \frac{\sum_{i=1}^{n_\theta} \theta_i \phi_i(\vec{x})}{\sum_{i=1}^{n_\theta} \phi_i(\vec{x})} \quad (6)$$

respectively.

- 3) Each basis function has the same width in each dimension, i.e., $\sigma_{i1} = \sigma_{i2} = \dots = \sigma_{in_{x_i}} = \sigma_i$. The activation function of each unit (3) then simplifies to

$$\begin{aligned} \phi_i(\vec{x}) &= \exp \left[- \left(\frac{(x_1 - c_{i1})^2}{\sigma_i^2} + \frac{(x_2 - c_{i2})^2}{\sigma_i^2} + \dots \right. \right. \\ &\quad \left. \left. + \frac{(x_{n_x} - c_{in_x})^2}{\sigma_i^2} \right) \right] \\ &= \exp \left[- \frac{\|\vec{x} - \vec{c}_i\|^2}{\sigma_i^2} \right] \end{aligned} \quad (7)$$

where $\|\cdot\|$ denotes the Euclidean norm.

It is this standard form of Gaussian RBF which is considered in the work of Jang and Sun [4].

III. THE TAKAGI–SUGENO MODEL OF FUZZY INFERENCE

A. The TS-Model

The Takagi–Sugeno model was introduced in [3] and [12] as a hybrid model, integrating both fuzzy conditions and

functional relationships between the input and output spaces. Rules have the canonical form

R_i : if x_{i1} is $A_{i1} \wedge x_{i2}$ is $A_{i2} \wedge \dots \wedge x_{in_{x_i}}$ is $A_{in_{x_i}}$ then $y = \theta_i(\vec{x})$,

with $i = 1 \dots n_r^r$, n_r^r being the number of rules. Each rule is premised on its own input vector \vec{x}_i where \vec{x}_i is a subset of the complete system input vector \vec{x} , i.e., $\vec{x}_i \subset \vec{x}$. A_{ij} are linguistic labels of fuzzy sets describing the qualitative state of the input variables, \wedge is a fuzzy conjunction operator (usually of T-Norm-characteristics) and the rule output y is a linear or nonlinear function [13] of the input variables. Rule inference is realized by first calculating the fulfillment, or firing strength, of the premise part as

$$\mu_{R_i}(\vec{x}_i) = \mu_{i1}(x_{i1}) \wedge \dots \wedge \mu_{in_{x_i}}(x_{in_{x_i}}).$$

Here, μ_{ij} is the membership function of fuzzy set A_{ij} . The firing strength is then multiplied with the output function, defining a locally valid model on the support of the Cartesian product of the fuzzy sets involved in building the premise. The fulfillment of the premise part can be calculated using multiplication or the minimum operator. The overall output of the TS-model is usually defined as the normalized sum (compare [14])

$$y = f(\vec{x}) = \frac{\sum_{i=1}^{n_r^r} \mu_{R_i}(\vec{x}_i) \cdot \theta_i(\vec{x})}{\sum_{i=1}^{n_r^r} \mu_{R_i}(\vec{x}_i)} \quad (8)$$

however, sometimes this is simplified to the unnormalized form

$$y = f(\vec{x}) = \sum_{i=1}^{n_r^r} \mu_{R_i}(\vec{x}_i) \cdot \theta_i(\vec{x}). \quad (9)$$

The TS-model has proved its applicability in various fields like controlling a model car [15], autopilot design and chemical process control [14], [13]. It is especially suitable for combining prototypical mathematical models with linguistically formulated conditions in the input space. The identification process of TS-type-models is very similar to the learning schemes of basis function nets. It consists of finding an adequate fuzzy partition of the input space (defining the shapes and number of the fuzzy sets, generating the rulebase) and fixing the parameters of the local models [14], [13].

B. Connections to Other Inference Mechanisms

The TS-model can be interpreted as an extension of some classical schemes of fuzzy inference. Consider the compositional rule of inference (CRI) proposed by Zadeh together with singleton fuzzification (represents sharp input values in the set Fuzzy(X) denoting all fuzzy sets defined over the domain X), sum-product composition. Using individual rule inference and defuzzification followed by additive superposition, results in an evaluation algorithm equivalent to a TS-system with constant local models. Tsukamoto's proposal [16] of fuzzy inference based on invertible output sets O_i can be readily modeled by a TS-system having nonlinear local models of the type

$$\theta_i(\vec{x}) = \mu_{O_i}^{-1}(\mu_{R_i}(\vec{x}_i)). \quad (10)$$

It is interesting to note that Kosko's additive fuzzy models denoted as BIOFAM (binary input output fuzzy associative memory) [17] can be embedded in the framework of the TS-model with only minor restrictions concerning the inference operators (algebraic product provides compatibility with the TS-inference) and the universe of discourse. To see this, let us consider the COG (center of gravity) defuzzification represented by the operator F_{COG}^{-1} : Fuzzy(Y) $\mapsto \mathcal{R}$ mapping every fuzzy output set $O_i \subset$ Fuzzy(Y), $i = 1, \dots, n_r^r$ to the real line making use of the COG-defuzzification scheme. After the inference procedure the membership functions of each rule's consequent part are transformed to μ'_{O_i} , where the prime operator denotes the restricted possibility distribution produced by the evaluation procedure. Kosko suggested to add the resulting membership functions instead of using a standard T-conorm operator like maximum. The application of the COG defuzzification to the sum $\mu'_O = \sum_{i=1}^{n_r^r} \mu'_{O_i}$ leads to the expression

$$\begin{aligned} & F_{COG}^{-1}(O'_1 + \dots + O'_{n_r^r}) \\ &= F_{COG}^{-1}\left(\sum_{i=1}^{n_r^r} \mu_{R_i}(\vec{x}_i) \cdot \mu_{O_i}\right) \\ &= \frac{\int_{D(y)} y \cdot \left(\sum_{i=1}^{n_r^r} \mu_{R_i}(\vec{x}_i) \cdot \mu_{O_i}\right) dy}{\int_{D(y)} \sum_{i=1}^{n_r^r} \mu_{R_i}(\vec{x}_i) \cdot \mu_{O_i} dy} \\ &= \frac{\sum_{i=1}^{n_r^r} \mu_{R_i}(\vec{x}_i) \int_{D(y)} y \cdot \mu_{O_i} dy}{\sum_{i=1}^{n_r^r} \mu_{R_i}(\vec{x}_i) \int_{D(y)} \mu_{O_i} dy} \\ &= \frac{\sum_{i=1}^{n_r^r} \mu_{R_i}(\vec{x}_i) \text{area}(O_i) F_{COG}^{-1}(O_i)}{\sum_{i=1}^{n_r^r} \mu_{R_i}(\vec{x}_i) \text{area}(O_i)}. \quad (11) \end{aligned}$$

Equation (11) can be reduced to a TS-type output with constant local models $\theta_i(\vec{x}) = F_{COG}^{-1}(O_i)$ by using fuzzy sets with $\text{area}(O_1) = \text{area}(O_2) = \dots = \text{area}(O_{n_r^r})$ or by using a scaling operation $G : Y \mapsto [0, 1]$ to normalize the domain Y and defining new premises $\mu_{\tilde{R}_i} = \text{area}(O_i) \cdot \mu_{R_i}$.

IV. FUNCTIONAL EQUIVALENCE

We are now in a position to establish the functional equivalence of the generalized RBF network and the TS-model of fuzzy inference. First we recall the result of Jang and Sun [4] which relates to the standard RBF network described by (5)–(7).

A. Jang-Sun Result

It is shown by Jang and Sun that the standard RBF network (5)–(7) is functionally equivalent to the TS-model of fuzzy inference under the following five conditions:

- 1) The number of RBF units is equal to the number of fuzzy if-then rules, i.e., $n_\theta = n_r^r$.
- 2) The output of each fuzzy if-then rule is a constant, i.e., $\theta_i(\vec{x}) = \theta_i$.
- 3) The membership functions within each rule are chosen as Gaussian functions with the same width.

- 4) The T -norm operator used to compute each rule's firing strength is multiplication.
- 5) Both the RBF network and the fuzzy inference system use the same method to derive their overall outputs, i.e., either the normalized or unnormalized calculation.

It seems clear that Conditions 1), 4), and 5) are natural conditions required to make the two systems structurally equivalent and that they cause no loss of generality in the type of RBF or fuzzy inference system which can be used. Conditions 2) and 3), on the other hand, significantly restrict the class of TS-model fuzzy inference systems to which this result applies. Our aim here is to remove Condition 2) and to remove the restriction in Condition 3) to functions with the same width.

B. Generalized Functional Equivalence

When we consider the generalized Gaussian RBF network defined by (1)–(4) the functional equivalence result can be generalized as follows:

Theorem 1: The generalized Gaussian RBF network defined by (1)–(4) is functionally equivalent to the TS-model of fuzzy inference if the following conditions are satisfied:

- 1) The number of RBF units is equal to the number of fuzzy if-then rules, i.e., $n_\theta = n_\theta^*$.
- 2) The membership functions within each rule are chosen as Gaussian functions.
- 3) The T -norm operator used to compute each rule's firing strength is multiplication.
- 4) Both the RBF network and the fuzzy inference system use the same method to derive their overall outputs, i.e., either the normalized or unnormalized calculation. \square

Proof: Under 2) each fuzzy if-then rule consists of the composition of the univariate Gaussian functions which define the membership values in the premise part of each rule. Each univariate membership function then has the form

$$\mu_{ij}(x_{ij}) = \exp \left[-\frac{(x_{ij} - c_{ij})^2}{\sigma_{ij}^2} \right], \quad j = 1 \cdots n_{x_i} \quad (12)$$

and this defines the j th membership value of the i th rule. Under 3) the firing strength of each rule is given by

$$\mu_{R_i}(\vec{x}_i) = \prod_{j=1}^{n_{x_i}} \mu_{ij}(x_{ij}) \quad (13)$$

for $i = 1 \cdots n_\theta$ [appealing to 1)]. This expands to

$$\begin{aligned} \mu_{R_i}(\vec{x}_i) &= \exp \left[-\left(\frac{(x_{i1} - c_{i1})^2}{\sigma_{i1}^2} + \frac{(x_{i2} - c_{i2})^2}{\sigma_{i2}^2} + \cdots \right. \right. \\ &\quad \left. \left. + \frac{(x_{in_{x_i}} - c_{in_{x_i}})^2}{\sigma_{in_{x_i}}^2} \right) \right] \\ &= \exp [-(\vec{x}_i - \vec{c}_i)' \Delta_i (\vec{x}_i - \vec{c}_i)] \quad (14) \end{aligned}$$

with

$$\Delta_i = \begin{bmatrix} \frac{1}{\sigma_{i1}^2} & 0 & \cdots \\ 0 & \frac{1}{\sigma_{i2}^2} & \\ \vdots & & \ddots \\ & & & \frac{1}{\sigma_{in_{x_i}}^2} \end{bmatrix}. \quad (15)$$

Comparing (14) and (15) with (3) and (4) the functional equivalence is established since $\mu_{R_i}(\vec{x}_i) = \phi_i(\vec{x}_i)$ and under this condition (together with $n_\theta^* = n_\theta$) (1) and (9) [or (2) and (8)] become identical. Necessity of Condition 4) is obvious, and means that either (1) and (9) are used together, or (2) and (8) are used together. It is clear that the firing strength of each rule $\mu_{R_i}(\vec{x}_i)$ functionally equates to the activation $\phi_i(\vec{x}_i)$ of the nonlinear processing unit in the corresponding RBF network. \square

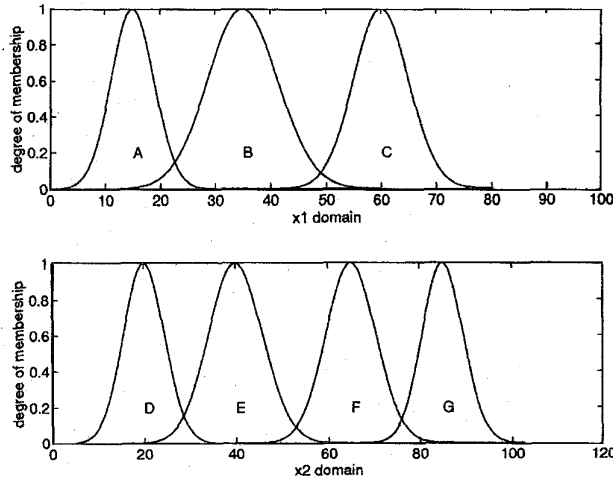
C. Discussion

Conditions 1), 3), and 4) in Theorem 1 are equivalent, respectively, to the Jang-Sun Conditions 1), 4), and 5). As mentioned already, these are natural conditions leading to no loss of generality.

The Jang-Sun Condition 2) on the other hand (output of each fuzzy rule must be a constant) has been eliminated. This is a direct result of the local model form of the generalized basis function network (1) or (2). Further, the local model network represents a generalization of the TS-model of fuzzy inference with respect to the original work of Takagi and Sugeno [14]. There, the output of each rule was given by a linear combination of elements of the input vector (corresponding to linear local models). In this paper the formulation in (1) and (2) allows generally nonlinear local models. Removal of Jang-Sun Condition 2) is therefore an important extension of the Jang-Sun functional equivalence result.

The Jang-Sun Condition 3) is related to our Condition 2). In contrast to Jang-Sun Condition 3), however, we require no restriction on the widths of the basis functions. This is a consequence of our employment of hyperellipsoidal basis functions (3) and (4) having differing widths in each dimension. This means that the univariate functions making up a specific rule (or RBF unit) are less interdependent. This is intuitively appealing as there is no practical reason why basis functions in different dimensions should be coupled by forcing them to have the same widths.

There is a final point which was not explicitly dealt with in the work of Jang and Sun. This relates to the fact that the premise part of each fuzzy if-then rule does not necessarily include conditions on every element of the input vector. Leaving out a specific element from the premise is equivalent to replacing the membership value for that element by one. This can only be achieved in a cumbersome way in the equivalent standard RBF formulation since the dimension of unit centers in the standard RBF is equal to the input dimension, i.e., $\dim(\vec{c}_i) = \dim(\vec{x}) = n_x$ in (7). To achieve a unity activation of the univariate basis function concerned requires adaptively setting the univariate center of the basis

Fig. 2. Membership functions $\mu_A(\cdot)$ – $\mu_G(\cdot)$.

function to the corresponding element of the input vector. For example, when the j th membership value of the i th rule is one (i.e., x_j is not conditioned in the premise) then we must set $c_{ij} = x_j$ in the equivalent standard RBF implementation for each new input vector. In contrast, the desired effect is inherent in the generalized RBF formulation of (1)–(4) since each unit in the network has as input only a subset \vec{x}_i of the input vector \vec{x} . It is clear to see that \vec{x}_i consists only of those elements of the input vector which are conditioned in the premise of rule i .

D. Example

A simple example should illustrate the results of this paper. Let us consider a fuzzy system of TS-type-inference with two-dimensional input $\vec{x} = (x_1, x_2)'$ and one-dimensional output y consisting of five rules:

- R_1 : if x_1 is $A \wedge x_2$ is E then $y = \theta_1(\vec{x})$
 R_2 : if x_1 is B then $y = \theta_2(\vec{x})$
 R_3 : if x_1 is $C \wedge x_2$ is F then $y = \theta_3(\vec{x})$
 R_4 : if x_1 is $C \wedge x_2$ is G then $y = \theta_4(\vec{x})$
 R_5 : if x_2 is D then $y = \theta_5(\vec{x})$.

The membership functions for the sets $A \cdots G$ are Gaussian shaped as shown in Fig. 2.

The centers and widths of the membership functions are

$$\begin{aligned} c_A &= 15, c_B = 35, c_C = 60, c_D = 20, \\ c_E &= 40, c_F = 65, c_G = 85, \\ \sigma_A^2 &= 30, \sigma_B^2 = 80, \sigma_C^2 = 50, \sigma_D^2 = 40, \\ \sigma_E^2 &= 70, \sigma_F^2 = 60, \sigma_G^2 = 40 \end{aligned}$$

where the subscripting arrangement is obvious.

The above rule base can be identified with the canonical rule base defined in Section III under the following conditions:

$$\vec{x}_1 = (x_1, x_2)', \vec{x}_2 = x_1, \vec{x}_3 = (x_1, x_2)', \vec{x}_4 = (x_1, x_2)', \vec{x}_5 = x_2$$

and

$$\vec{A}_1 = (A, E)', \vec{A}_2 = B, \vec{A}_3 = (C, F)', \vec{A}_4 = (C, G)', \vec{A}_5 = D.$$

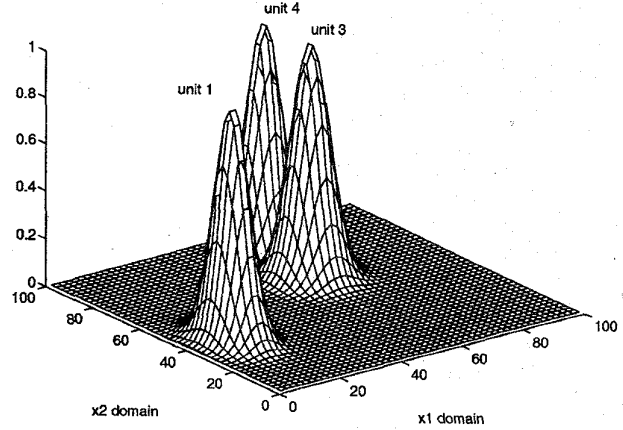


Fig. 3. RBF basis functions—units 1, 3, and 4.

Based on the theorem developed in the last section, the mapping of the fuzzy rulebase to a functionally equivalent RBF-structure is straightforward. The five RBF units ($n_\theta = n_r^r = 5$) are defined to be equivalent to the fuzzy rule firing strengths

$$\phi_i(\vec{x}_i) = \mu_{R_i}(\vec{x}_i) \quad (16)$$

for $i = 1 \cdots 5$. Specifically, we have

$$\begin{aligned} \phi_1(\vec{x}_1) &= \mu_A(x_1)\mu_E(x_2) \\ \phi_2(\vec{x}_2) &= \mu_B(x_1) \\ \phi_3(\vec{x}_3) &= \mu_C(x_1)\mu_F(x_2) \\ \phi_4(\vec{x}_4) &= \mu_C(x_1)\mu_G(x_2) \\ \phi_5(\vec{x}_5) &= \mu_D(x_2). \end{aligned}$$

The centers and widths of these RBF basis functions are defined directly by the parameters of the fuzzy membership functions as stated above. They are

$$\begin{aligned} \vec{c}_1 &= (c_A, c_E)' = (15, 40)', \quad \vec{c}_2 = c_B = 35, \\ \vec{c}_3 &= (c_C, c_F)' = (60, 65)', \\ \vec{c}_4 &= (c_C, c_G)' = (60, 85)', \quad \vec{c}_5 = c_D = 20, \end{aligned}$$

$$\Delta_1 = \begin{bmatrix} \frac{1}{\sigma_A^2} & 0 \\ 0 & \frac{1}{\sigma_E^2} \end{bmatrix} = \begin{bmatrix} \frac{1}{30} & 0 \\ 0 & \frac{1}{70} \end{bmatrix},$$

$$\Delta_2 = \frac{1}{\sigma_B^2} = \frac{1}{80}, \quad \Delta_3 = \begin{bmatrix} \frac{1}{\sigma_C^2} & 0 \\ 0 & \frac{1}{\sigma_F^2} \end{bmatrix} = \begin{bmatrix} \frac{1}{50} & 0 \\ 0 & \frac{1}{60} \end{bmatrix},$$

$$\Delta_4 = \begin{bmatrix} \frac{1}{\sigma_C^2} & 0 \\ 0 & \frac{1}{\sigma_G^2} \end{bmatrix} = \begin{bmatrix} \frac{1}{50} & 0 \\ 0 & \frac{1}{40} \end{bmatrix}, \quad \Delta_5 = \frac{1}{\sigma_D^2} = \frac{1}{40}.$$

The ellipsoidal basis functions for units 1, 3, and 4 are plotted in Fig. 3. The basis functions for units 2 and 5 can be seen in Fig. 4.

The functional relationships defined by the rule consequences can be directly interpreted in the RBF-context as local

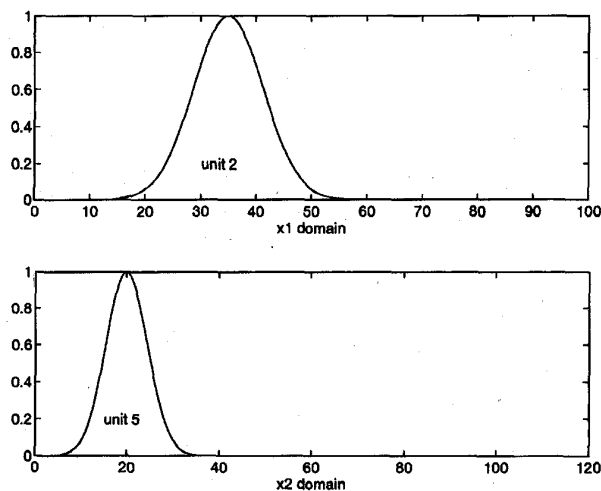


Fig. 4. RBF's—units 2 and 5.

models $\theta_i(\vec{x})$ resulting in the functionally equivalent outputs (unnormalized case)

$$f_{\text{RBF}}(\vec{x}) = \sum_{i=1}^5 \phi_i(\vec{x}_i)\theta_i(\vec{x}) = \sum_{i=1}^5 \mu_{R_i}(\vec{x}_i)\theta_i(\vec{x}) = f_{\text{TS}}(\vec{x}). \quad (17)$$

V. CONCLUSIONS

The primary purpose of this paper was the extension of the equivalence relation between a class of TS-type fuzzy systems with Gaussian membership functions and RBF networks first established by Jang and Sun [4], and rooted in the fundamental isomorphy property of the exponential function mapping the group $(\mathcal{R}, +)$ to the group $(\mathcal{R} \setminus \{0\}, \cdot)$. We proved that the restrictions concerning the variances of the membership functions and the use of constant weights can be easily removed in the framework of generalized RBF nets with local models and discussed the problem of dealing with rule premises that are not defined on the full input space. It should be emphasized that the equivalence theorem holds not only for the TS-model of fuzzy inference but for a quite large class of systems which can be transformed to the TS-model, including a subset of Kosko's popular BIOFAM systems. The equivalence theorem has immediate application for the following:

- integration of linguistically structured *a priori* knowledge into RBF networks;

- increase of the cognitive transparency of RBF networks;
- transfer of training and learning algorithms; and
- development of hybrid learning techniques mixing symbolic and connectionist approaches.

While some aspects have already been used implicitly in numerous works on hybrid learning and approximation theoretic analysis, the main advantage of functional equivalence is the provision of a unified framework for the analysis of basis function nets, local model nets, and fuzzy systems.

REFERENCES

- [1] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Syst.*, vol. 2, pp. 321–355, 1988.
- [2] J. Moody and C. Darken, "Fast-learning in networks of locally-tuned processing units," *Neural Comput.*, vol. 1, pp. 281–294, 1989.
- [3] T. Takagi and M. Sugeno, "A new approach to the design of fuzzy controller," in *Advances in Fuzzy Sets, Possibility Theory, and Applications*, P. Wang and S. Chang, Eds., New York: Plenum, 1983, pp. 325–334.
- [4] J. S. R. Jang and C. T. Sun, "Functional equivalence between radial basis function networks and fuzzy inference systems," *IEEE Trans. Neural Networks*, vol. 4, pp. 156–158, Jan. 1993.
- [5] R. D. Jones, Y. C. Lee, C. W. Barnes, G. W. Flake, K. Lee, S. P. Lewis, and S. Qian, "Function approximation and time-series prediction with neural networks," in *Proc. IEEE Conf. Neural Networks*, 1989.
- [6] T. A. Johansen and B. A. Foss, "Representing and learning unmodeled dynamics with neural network memories," in *Proc. Amer. Contr. Conf.*, Chicago, IL, June 1992, pp. 3037–3043.
- [7] B. A. Foss and T. A. Johansen, "On local and fuzzy modeling," in *Proc. 3rd Int. Conf. Ind. Fuzzy Contr. Intell. Syst.*, Houston, TX, 1993.
- [8] K. J. Hunt, R. Haas, and M. Brown, "On the functional equivalence of fuzzy inference systems and spline-based networks," *Int. J. Neural Syst.*, June 1995.
- [9] R. Shorten and R. Murray-Smith, "On normalizing basis function networks," in *Proc. 4th Irish Neural Networks Conf.*, Univ. College Dublin, Sept. 1994.
- [10] T. A. Johansen, "Operating Regime Based Process Modeling and Identification," Ph.D. dissertation, Norges Tekniske Høgskole, Trondheim, Norway, Nov. 1994.
- [11] R. Murray-Smith, "A Local Model Network Approach to Nonlinear Modeling," Ph.D. dissertation, Dep. Comput. Sci., Univ. Strathclyde, Glasgow, Scotland, Nov. 1994.
- [12] T. Takagi and M. Sugeno, "Derivation of fuzzy control rules from human operator's control actions," in *Proc. IFAC Symp. Fuzzy Inform. Knowledge Representation Decision Anal.*, 1983, pp. 55–60.
- [13] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Trans. Fuzzy Syst.*, pp. 7–31, Feb. 1993.
- [14] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications for modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. 15, no. 1, pp. 116–132, 1985.
- [15] M. Sugeno and M. Nishida, "Fuzzy control of a model car," *Fuzzy Sets Syst.*, vol. 16, pp. 103–118, 1985.
- [16] Y. Tsukamoto, "An approach to fuzzy reasoning," in *Advances in Fuzzy Set Theory and Applications*, M. M. Gupta, R. K. Ragade, and R. R. Yager, Eds., Amsterdam, The Netherlands: North Holland, 1979.
- [17] B. Kosko, *Neural Nets and Fuzzy Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1992.