Control and Cybernetics

vol. **31** (2002) No. 2

A neuro – fuzzy controller with a compromise fuzzy reasoning

by

L. Rutkowski and K. Cpałka

Department of Computer Engineering,
Technical University of Częstochowa,
Częstochowa, Poland

Abstract: This paper presents a compromise approach to neuro – fuzzy controllers. It includes both Mamdani (constructive) and logical (destructive) fuzzy inference. New neuro – fuzzy controllers are derived and simulation results are presented.

Keywords: neuro – fuzzy controler, Mamdani approach, logical approach.

1. Introduction

It is well known that the fuzzy–logic control does not require a conventional model of the process, contrary to the classical control techniques, which are based on analytical or experimental models. Moreover, traditional controllers cannot incorporate the linguistic fuzzy information – coming from human experts - into their design. On the other hand, fuzzy controllers suffer from the lack of learning properties. Therefore, several neuro – fuzzy controllers have been developed (see, e.g., Cpałka, 2001, Cpałka and Rutkowski, 2000, Jang, Sun and Mizutani, 1997, Rutkowska, 2001, Rutkowska and Nowicki, 2000, Rutkowska, Piliński and Rutkowski, 1997, Rutkowski nad Cpałka, 2000, 2001, Wang, 1994, Yager and Filev, 1994). They exhibit advantages of neural networks and fuzzy systems. In particular, fuzzy – neural controllers combine learning abilities of neural networks and natural language description of fuzzy systems.

The structure of such controllers depends on implications used in fuzzy systems: Mamdani–type implication

$$I_1(a,b) = \min\{a,b\} \tag{1}$$

or logical–type implication, e.g. binary implication

$$I_2(a,b) = \max\{1-a,b\}. \tag{2}$$

Strictly speaking, formula (1) does not satisfy the definition of fuzzy implication

Historically, an application of formula (1) to fuzzy control systems was first reported in Mamdani and Assilian (1975). Consequently, the existing neuro–fuzzy controllers employ two different approaches and the literature offers no suggestions as to which of them is superior. Therefore, in this paper we combine both approaches and present a compromise approach to neuro–fuzzy control systems design. We propose a soft fuzzy implication given by

$$I(a,b) = (1-\nu)I_1(a,b) + \nu I_2(a,b) \tag{3}$$

where $\nu \in [0,1]$, and based on implication (3) we derive a compromise neuro–fuzzy controller. It includes Mamdani–type, logical–type, more Mamdani–type than logical–type and more logical–type than Mamdani–type fuzzy inference systems. It should be emphasized that our neuro–fuzzy controller can be optimised with respect to the parameters of fuzzy membership functions in the process of learning. Moreover, also the parameter $\nu \in [0,1]$, determining the type of the system, can be found in the process of learning. To our best knowledge such result has not been presented in the literature yet. The compromise neuro–fuzzy controllers developed in the paper are simulated on the truck backer–upper control problem. In the sequel S, T and N denote S-norm, T-norm and negation, respectively. By making use of this notation formula (3) can be generalized to the form

$$I(a,b) = N(\nu)T\{a,b\} + \nu S\{N(a),b\} \tag{4}$$

where $N(\nu) = 1-\nu$.

2. Mamdani and logical fuzzy inference controllers

In this paper, we consider multi–input, single–output fuzzy controllers mapping $\mathbf{X} \rightarrow \mathbf{Y}$, where $\mathbf{X} \subset \mathbf{R}^n$ and $\mathbf{Y} \subset \mathbf{R}$, see Fig. 1.

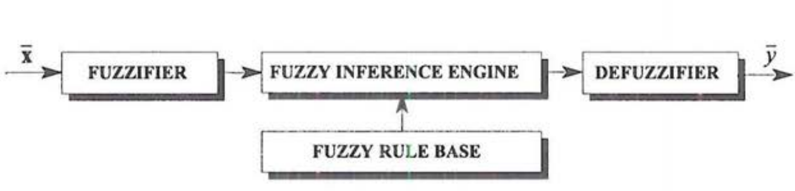


Figure 1. Fuzzy–logic controller

The fuzzifier performs a mapping from the observed crisp input space $\mathbf{X} \subset \mathbf{R}^n$

fuzzifier which maps $\overline{\mathbf{x}} = [\overline{x}_1, \ldots, \overline{x}_n] \in \mathbf{X}$ into a fuzzy set $A' \subset \mathbf{X}$ characterized by the membership function

$$\mu_{A'}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} = \overline{\mathbf{x}} \\ 0 & \text{if } \mathbf{x} \neq \overline{\mathbf{x}} \end{cases}. \tag{5}$$

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

$$R^{(k)}: \text{ IF } x_1 \text{ is } A_1^k \text{ AND } x_2 \text{ is } A_2^k \text{ AND} \ldots \text{AND } x_n \text{ is } A_n^k \text{ THEN } y \text{ is} B^k \tag{6}$$

or

$$R^{(k)}: \text{IF } \mathbf{x} \text{ is } A^k \text{ THEN } y \text{ is } B^k \tag{7}$$

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

The fuzzy inference determines a mapping from the fuzzy sets in the input space $\mathbf{X}$ to the fuzzy sets in the output space $\mathbf{Y}$. Each of N rules (7) determines a fuzzy set $\overline{B}^k \subset \mathbf{Y}$ given by the compositional rule of inference

$$\overline{B}^k = A' \circ (A^k \to B^k) \tag{8}$$

where $A^k = A_1^k \times A_2^k \times \ldots \times A_n^k$. Fuzzy sets $\overline{B}^k$, according to the formula (8), are characterized by membership functions expressed by the sup-star composition

$$\mu_{\overline{B}^k}(y) = \sup_{\mathbf{x} \in \mathbf{X}} \{\mu_{A'}(\mathbf{x}) * \mu_{A_1^k \times \ldots \times A_n^k \to B^k}(\mathbf{x}, y)\} \tag{9}$$

where $*$ can be any operator in the class of T-norms. It can be easily seen that for a crisp input $\overline{\mathbf{x}} \in \mathbf{X}$, i.e. a singleton fuzzifier (5), formula (9) becomes

$$\mu_{\overline{B}^k}(y) = \mu_{A_1^k \times \ldots \times A_n^k \to B^k}(\overline{\mathbf{x}}, y) = \mu_{A^k \to B^k}(\overline{\mathbf{x}}, y) = I(\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(y)) \tag{10}$$

where $I(\cdot)$ is a fuzzy implication. The aggregation operator, applied in order to obtain the fuzzy set B' based on fuzzy sets $\overline{B}^k$, is the T-norm or S-norm operator, depending on the type of fuzzy implication.

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

$$\overline{y} = \frac{\int_{\mathbf{Y}} y \mu_{B'}(y)\, dy}{\int_{\mathbf{Y}} \mu_{B'}(y)\, dy} \tag{11}$$

or by

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot \mu_{B'}(\overline{y}^r)}{} \tag{12}$$

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

$$\mu_{B^r}(\overline{y}^r) = \max_{y \in \mathbf{Y}} \{\mu_{B^r}(y)\}. \tag{13}$$

a) Mamdani's approach

In this approach, fuzzy implication (10) is a T-norm (e.g. minimum or product)

$$I(\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)) = T\{\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)\} \tag{14}$$

and the aggregated output fuzzy set $B' \subset \mathbf{Y}$ is given by

$$\mu_{B'}(\overline{y}^r) = \mathop{S}_{k=1}^{N} \{\mu_{B^r}(\overline{y}^r)\} = \mathop{S}_{k=1}^{N} \{T\{\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)\}\}. \tag{15}$$

Consequently, formula (12) takes the form

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot S_{k=1}^{N} \{T\{T_{i=1}^{n}\{\mu_{A_i^k}(\overline{x}_i)\}, \mu_{B^k}(\overline{y}^r)\}\}}{\sum_{r=1}^{N} S_{k=1}^{N} \{T\{T_{i=1}^{n}\{\mu_{A_i^k}(\overline{x}_i)\}, \mu_{B^k}(\overline{y}^r)\}\}} \tag{16}$$

b) Logical approach

In this approach the fuzzy implication (10) is an S-implication in the form

$$I(\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)) = S\{N(\mu_{A^k}(\overline{\mathbf{x}})), \mu_{B^k}(\overline{y}^r)\} \tag{17}$$

e.g. binary implication (known as the Kleene – Dienes implication)

$$I(\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)) = \max\{1 - \mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)\}. \tag{18}$$

The aggregated output fuzzy set $B' \subset \mathbf{Y}$ is given by

$$\mu_{B'}(\overline{y}^r) = \mathop{T}_{k=1}^{N} \{\mu_{B^r}(\overline{y}^r)\} = \mathop{T}_{k=1}^{N} \{S\{N(\mu_{A^k}(\overline{\mathbf{x}})), \mu_{B^k}(\overline{y}^r)\}\} \tag{19}$$

and formula (12) becomes

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot T_{k=1}^{N} \{S\{N(T_{i=1}^{n}\{\mu_{A_i^k}(\overline{x}_i)\}), \mu_{B^k}(\overline{y}^r)\}\}}{\sum_{r=1}^{N} T_{k=1}^{N} \{S\{N(T_{i=1}^{n}\{\mu_{A_i^k}(\overline{x}_i)\}), \mu_{B^k}(\overline{y}^r)\}\}}. \tag{20}$$

In the next section we generalize both approaches described in the previous

3. The compromise neuro–fuzzy controller

It can be easily seen that the controllers (16) and (20) can be presented in the form

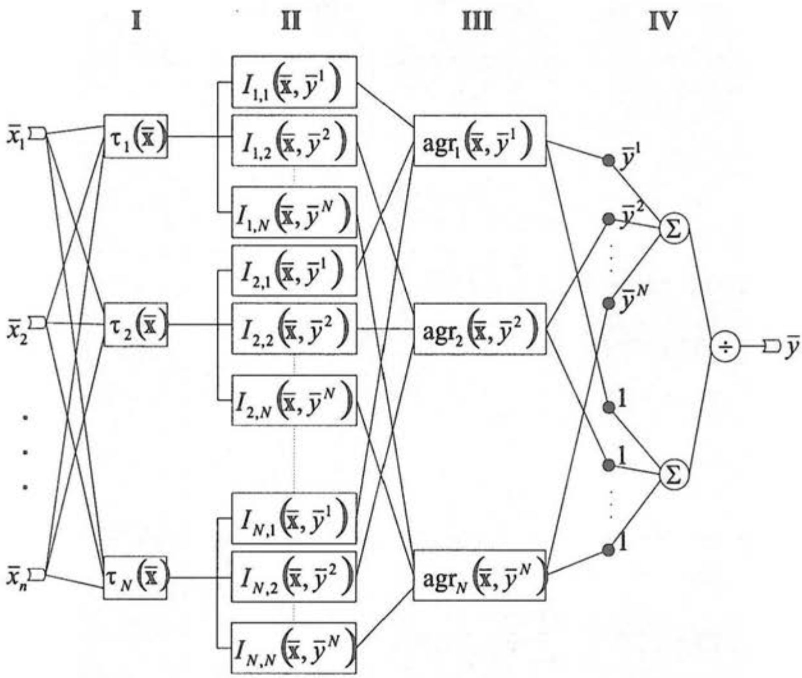
$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot \operatorname{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)}{\sum_{r=1}^{N} \operatorname{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)} \tag{21}$$

where

$$\operatorname{agr}_r(\overline{\mathbf{x}}, \overline{y}^r) = \begin{cases} \underset{k=1}{\overset{N}{S}} \{I_{k,r}(\overline{\mathbf{x}}, \overline{y}^r)\} & \text{for Mamdani's approach} \\ \underset{k=1}{\overset{N}{T}} \{I_{k,r}(\overline{\mathbf{x}}, \overline{y}^r)\} & \text{for logical approach} \end{cases} \tag{22}$$

and

$$I_{k,r}(\overline{\mathbf{x}}, \overline{y}^r) = \begin{cases} T\{\tau_k(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)\} & \text{for Mamdani's approach} \\ S\{N(\tau_k(\overline{\mathbf{x}})), \mu_{B^k}(\overline{y}^r)\} & \text{for logical approach,} \end{cases} . \tag{23}$$



Moreover, the firing strength of rules is given by

$$\tau_k(\overline{\mathbf{x}}) = \mathop{T}_{i=1}^{n} \{\mu_{A_i^k}(\overline{x}_i)\}. \tag{24}$$

The general architecture of the controller is depicted in Fig. 2. Observe that this architecture has a multilayer structure and can be trained by the back–propagation method.

The compromise neuro–fuzzy controller is based on formulas (22)–(24). We define this controller as follows:

$$\tau_k(\overline{\mathbf{x}}) = T\{\mu_{A_1^k}(\overline{x}_1), \ldots, \mu_{A_n^k}(\overline{x}_n)\} \tag{25}$$

$$I_{k,r}(\overline{\mathbf{x}}, \overline{y}^r) = \begin{pmatrix} N(\nu)T\{\tau_k(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)\}+ \\ +\nu S\{N(\tau_k(\overline{\mathbf{x}})), \mu_{B^k}(\overline{y}^r)\} \end{pmatrix} \tag{26}$$

$$\mathrm{agr}_r(\overline{\mathbf{x}}, \overline{y}^r) = \begin{pmatrix} N(\nu)S\{I_{1,r}(\overline{\mathbf{x}}, \overline{y}^r), \ldots, I_{N,r}(\overline{\mathbf{x}}, \overline{y}^r)\}+ \\ +\nu T\{I_{1,r}(\overline{\mathbf{x}}, \overline{y}^r), \ldots, I_{N,r}(\overline{\mathbf{x}}, \overline{y}^r)\} \end{pmatrix} \tag{27}$$

where $\nu \in [0,1]$.

4. Optimisation of fuzzy controller (21)

Observe that controller (21) is Mamdani–type for $\nu = 0$, more Mamdani–type than logical type for $\nu \in (0, 0.5)$, undetermined for $\nu = 0.5$, more logical–type than Mamdani–type for $\nu \in (0.5, 1)$, and logical type for $\nu = 1$. The 3D plots of this controller are shown in Fig. 3.

It is worth noticing that the parameter ν and, membership function parameters of A_i^k and B^k, $i = 1, \ldots, n$, $k = 1, \ldots, N$, can be learned. Consequently, type of the controller can be determined in the process of learning (see Section 5). Based on the learning sequence

$$(\overline{\mathbf{x}}(1), d(1)), (\overline{\mathbf{x}}(2), d(2)), \ldots \tag{28}$$

we optimise the index

$$Q = \frac{1}{2}(\overline{y} - d)^2 \tag{29}$$

with respect to parameter ν subject to the constraint $\nu \in [0,1]$. A standard steepest descent recursive procedure,

$$\nu(t+1) = \nu(t) - \eta \frac{\partial Q(t)}{\partial \nu(t)} \tag{30}$$

where $\eta \in [0,1]$ is the learning rate, is applied. For derivations and details the reader is referred to Cpałka (2001). The learning procedures are applied in the

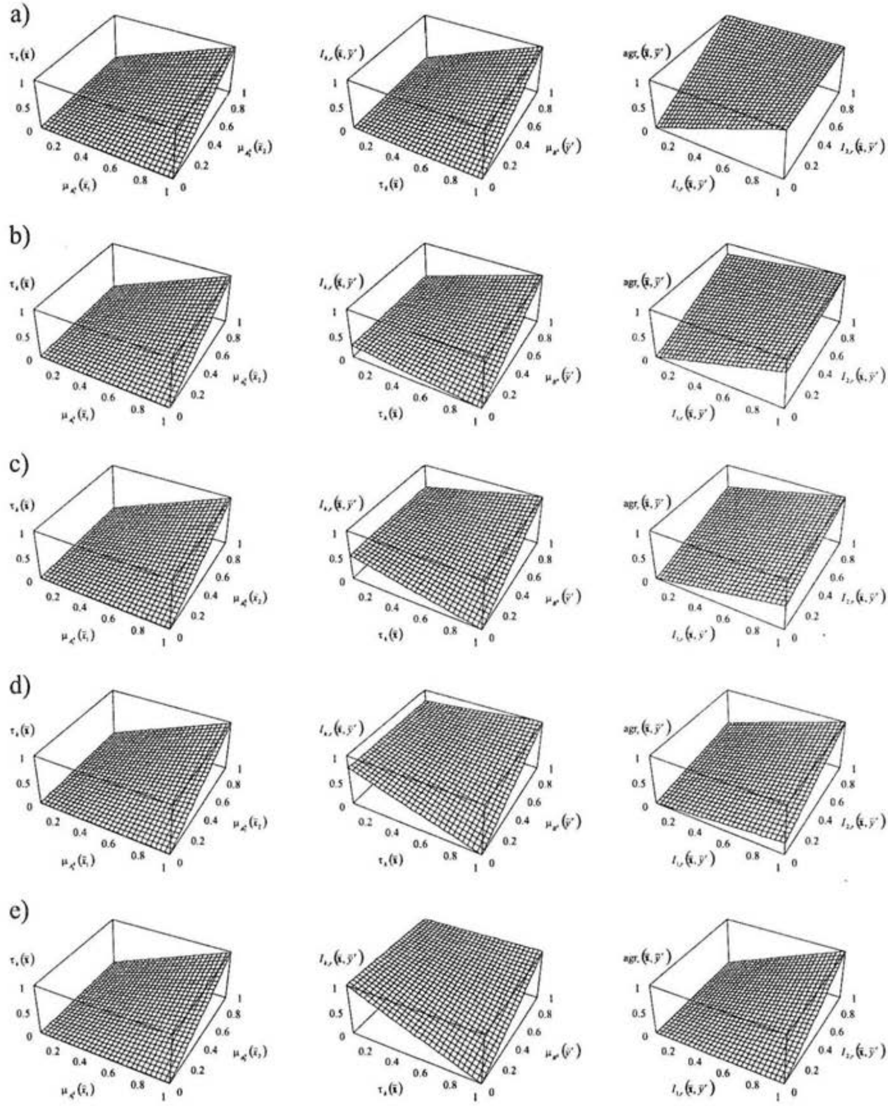


Figure 3. 3D plots of a compromise neuro–fuzzy controller given by (25)–(27) for $n = 2$, $N = 2$, a) $\nu = 0.00$, b) $\nu = 0.25$, c) $\nu = 0.50$, d) $\nu = 0.75$, e) $\nu = 1.00$

5. Application to truck backer–upper control

The compromise neuro–fuzzy controllers described by formulas (25)–(27) are simulated on the truck backer–upper control problem (Rutkowska, Piliński and Rutkowski, 1997)

a) Problem formulation

Fig. 4 shows the truck and its loading zone. The truck position is exactly determined by three state variables $x \in [-150, 150]$, $y \in [0, 300]$, $\phi \in [-180°, 180°]$, where ϕ is the angle between the truck and the vertical. Control of the truck is the steering angle $\phi \in [-45°, 45°]$. The truck moves backward by a fixed unit distance at every stage. For simplicity, we assume that there exists enough clearance between the truck and the loading dock so we can ignore the y - position coordinate. The goal is to design a fuzzy inference controller making the truck arrive at the loading dock at the angle $\phi(t_f) = 0$ and the final position $x(t_f) = 0$.

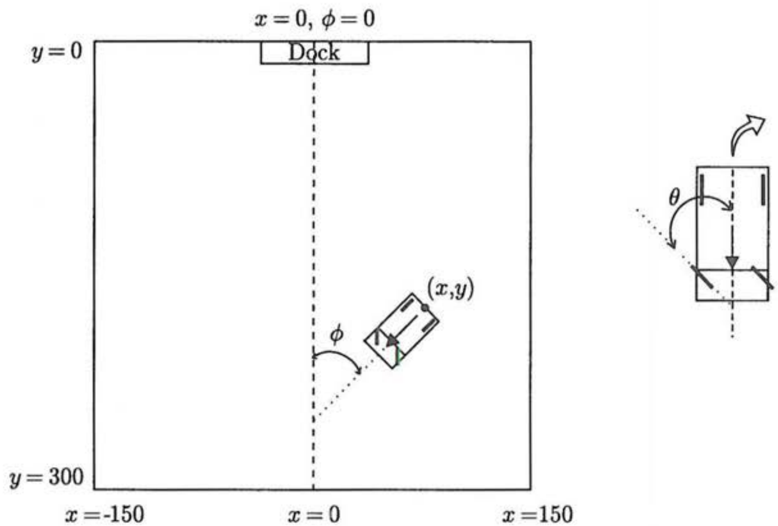


Figure 4. Diagram of simulated and loading zone

b) Generation of the learning sequences

We describe the movement of the truck by making use of the following approximate kinematics derived by Wang (1994):

$$x(t+1) = x(t) + \sin(\theta(t) + \phi(t)) - \sin(\theta(t))\cos(\phi(t)) \tag{31}$$

$$\phi(t+1) = \phi(t) - \arcsin\left(\frac{2}{b}\sin(\theta(t))\right) \tag{32}$$

where b is the length of the truck. In the simulation $b = 20$.

Based on the above equations we generate 14 learning sequences starting from different initial states $(x(t_0), \phi(t_0))$. The steering angle θ at every stage is chosen by a trial-and-error method such that the kinematics equations finally give $x(t_f) \approx 0$, $\phi(t_f) \approx 0$. The fourteen learning sequences of different lengths form one epoch of length 282.

c) Simulation result

We used the compromise neuro-fuzzy controller with probabilistic triangular norms, Gaussian membership functions and nine rules. The initial value of parameter ν was set to 0.5.

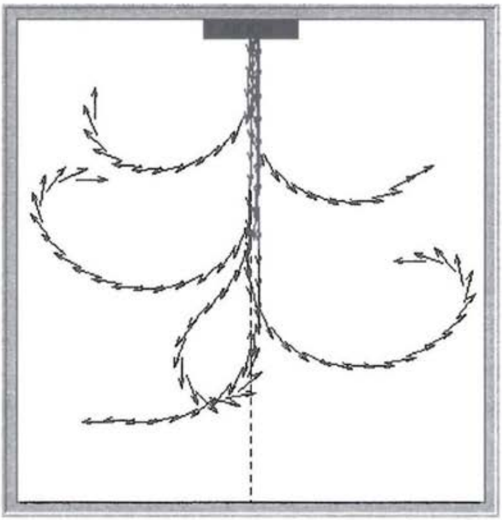


Figure 5. Truck trajectories

In the learning process the parameter reached the value equal 1 and the controller became of a logical type. The truck trajectories are shown in Fig. 5. We observe that the compromise neuro-fuzzy controller (21) successfully controls the truck to the desired position.

6. Application to modeling a two-input sinc function

In this example, we use compromise neuro-fuzzy controllers, described by formulas (25)–(27), to model a two-dimensional function (Jang, Sun and Mizutani,

a) Problem formulation and generation of the learning sequences
In this subsection, double–input and single output static function is chosen to be a target system for the new fuzzy modeling strategy. This function is represented as

$$y = \frac{\sin(x_1)\sin(x_2)}{x_1 x_2}. \tag{33}$$

From the evenly distributed grid of points of the input range $[-10, 10] \times [-10, 10]$ of equation (33), 121 training data pairs were obtained. The training data are shown in Fig. 6.

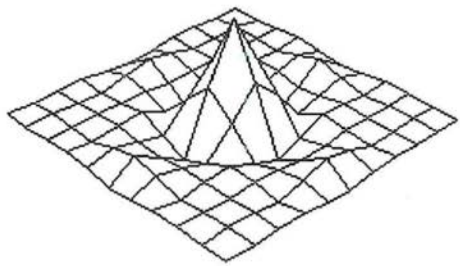
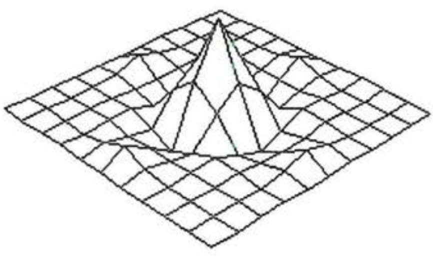


Figure 6. Training data

b) Simulation results
We used the compromise neuro–fuzzy controller with probabilistic triangular norms, Gaussian membership functions and nine rules. The initial value of parameter ν was set to 0.5.

In the learning process the parameter ν reached the value equal 0 and the controller become of a Mamdani type. A reconstructed surface is shown in Fig. 7. We observe that the compromise neuro–fuzzy controller (21) successfully approximates the nonlinear function. Our result is comparable with that of Jang, Sun and Mizutani (1997).



In Fig. 8 we depicted the surface obtained when a logical inference is applied. It can be easily seen that the reconstruction is not as good as shown in Fig. 7 for a Mamdani type inference.

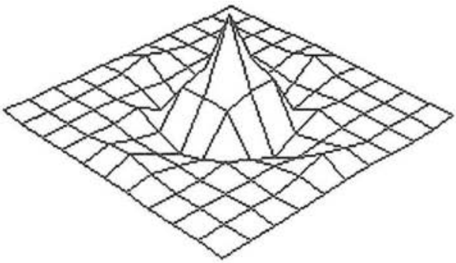


Figure 8. The reconstructed surface – the logical approach

7. Final remarks

In this paper we derived flexible structures of neuro–fuzzy controllers characterized by fuzzy implication (3) or (4). The parameter ν describing a type of the system is determined in the process of learning. It should be noted that the behaviour of the system does not depend on the initial value of the parameter ν. In Sections 5 and 6 we obtained the same final values of ν starting from $\nu(0) = 0.25; 0.50; 0.75$. Our approach introduces more flexibility into the structure of neuro–fuzzy controllers and significantly reduces the design effort.

References

CPAŁKA, K. (2001) *Flexible Neuro–Fuzzy Systems. Ph. D. dissertation*, Czestochowa.

CPAŁKA, K. and RUTKOWSKI, L. (2000) Soft neuro–fuzzy systems. *Proceedings of the Fifth Conference Neural Networks and Soft Computing*, Zakopane, 296–301.

FODOR, J. C. (1991) On fuzzy implication operators. *Fuzzy Sets and Systems*, **42**, 293–300.

JANG, J. S. R., SUN, C. T. and MIZUTANI, E. (1997) *Neuro–Fuzzy and Soft Computing*. Prentice Hall, Englewood Cliffs.

MAMDANI, E. H. and ASSILIAN, S. (1975) An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man – Machine Studies*, **7**, (1), 1–13.

MENDEL, J. M. (1995) Fuzzy logic systems for engineering: a tutorial. *IEEE*

RUTKOWSKA, D. (2001) *Neuro-Fuzzy Architectures and Hybrid Learning.* Springer-Verlag.

RUTKOWSKA, D. and NOWICKI, R. (2000) Implication-based neuro-fuzzy architectures. *Int. J. Appl. Math. Comput. Sci.*, **10**, 4, 675–701.

RUTKOWSKA, D., PILIŃSKI, M. and RUTKOWSKI, L. (1997) *Neural Networks, Genetic Algorithms and Fuzzy Systems.* PWN, Warszawa, in Polish.

RUTKOWSKI, L. and CPAŁKA, K. (2001) A general approach to neuro-fuzzy systems. *Proceedings of the 10th IEEE International Conference on Fuzzy Systems*, Melbourne.

RUTKOWSKI, L. and CPAŁKA, K. (2000) *Flexible structures of neuro-fuzzy systems.* Volume 54 of *Quo Vadis Computational Intelligence. "Studies in Fuzziness and Soft Computing"*, Springer-Verlag, 479–484.

WANG, L. X. (1994) *Adaptive Fuzzy Systems and Control.* Prentice Hall, Englewood Cliffs.

YAGER, R. R. and FILEV, D. P. (1994) *Essentials of Fuzzy Modelling and Control.* John Wiley & Sons, New York.