Appl. Math. and Comp. Sci., 1998, Vol.8, No.3, 611–632

NONREGULAR NONLINEAR SECTOR MODELLING

ANDRZEJ PIEGAT*

Fuzzy models realize human-like modelling schemes. However, a human being can create in his mind relatively simple models of real systems with maximum two inputs. The reason is that human models are based on rectangular lattice partitions of the input spaces. Such a partition enables us to understand the modelled system, which is a great advantage of fuzzy modelling. Despite this, the rectangular lattice partition makes modelling of systems with large numbers of inputs and those realizing complicated inputs/output mappings impossible or very difficult. The paper puts forward a self-organizing and self-tuning method for modelling nonlinear systems. It is based on a nonrectangular partition of the input space. The conclusions of rules can be here linear or nonlinear. For the latter, a special delinearization function (SDL) is proposed. It makes it possible to decrease considerably the number of rules, which results in efficient modelling. Also, the amount of measurement information from the system needed to learn a model can be decreased considerably.

1. Introduction

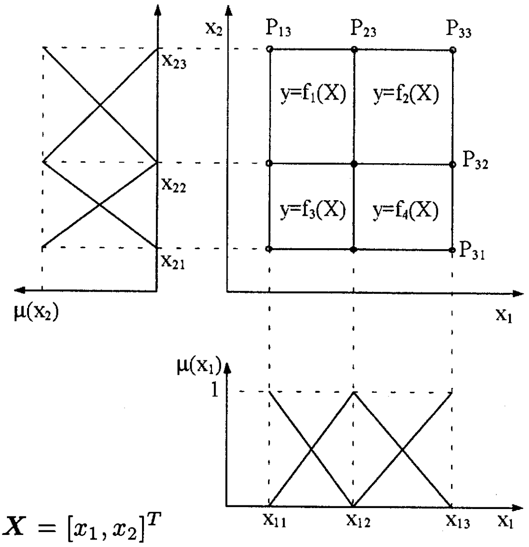
Zeng and Sing (1996) prove that any fuzzy system with commonly-used triangular membership functions is a piecewise multilinear function. "Pieces" in the case of 2D input-space rectangles creating a partition lattice are shown in Fig. 1. The output value y of the Mamdani model at each node P_{ij} of the partition lattice and in its neighbourhood is defined by a rule of the type

$$\text{IF } (x_1 \text{ close to } x_{1i}) \text{ AND } (x_2 \text{ close to } x_{2j}), \text{ THEN } (y \text{ close to } y_{ij}) \tag{1}$$

for $i, j = 1, 2, 3$.

If the statements like 'x_1 close to x_{1i}' are replaced by 'x = small', 'x = mean' and 'x = large', then the rules become easy to understand and to remember by a man who will create in his mind a fuzzy system model based on linguistic rules. In the example shown in Fig. 1, the model consists of nine rules which define the output value y for the input vector $\boldsymbol{X}$ and the neighbourhood of nine nodes P_{ij} of the partition lattice.

* Institute of Computer Science and Information Systems, Technical University of Szczecin, ul. Żołnierska 49, 71–210 Szczecin, Poland, e-mail: andrzej_piegat@ii.tuniv.szczecin.pl.



$$X = [x_1, x_2]^T$$

$$y = f_k(X) = a_{0k} + a_{1k}x_1 + a_{2k}x_2 + a_{3k}x_1x_2$$

Fig. 1. Rectangular partition of the input space in a fuzzy Mamdani model.

According to Zeng and Singh, instead of the rules (1), the following ones which describe the model surface over each sector of the partition lattice can be used:

$$\text{IF } (x_{1i} \le x_1 \le x_{1(i+1)}) \text{ AND } (x_{2j} \le x_2 \le x_{2(j+1)}), \text{ THEN } \big(y = f_k(x_1, x_2)\big) \quad (2)$$

where

$$y = f_k(x_1, x_2) = a_{0k} + a_{1k}x_1 + a_{2k}x_2 + a_{3k}x_1x_2 \quad (3)$$

is a multilinear function.

A very important equivalence condition of the model with point rules of type (1) and that with sector rules of type (2) is the identity of neighbouring functions f_k and f_{k+1} along the sector borders. If multilinear functions (3) are in the sector rules conclusions, then this condition is satisfied if the neighbouring functions f_k and f_{k+1} have identical values at their common nodes P_{ij}, because the borders of these multilinear functions are linear, which can be tested by imposing constant values on x_1 or x_2 in (3).

Application of sector rules (2) in the example shown in Fig. 1 decreases the number of rules from nine (the Mamdani model) to four (the sector model). However, the amount of measurement information needed to tune both the models is identical and equal to $r = m_1 \times m_2$, where m_i stands for the number of fuzzy sets of the input x_i, $i = 1, 2$. This number is equal to the number of nodes of the partition lattice. In the case of an n-dimensional input space the number of nodes of the rectangular lattice ($r = m_1 \times m_2 \times \cdots \times m_n$) increases sharply. For instance, for ten inputs, each defined by three linguistic values, we have $r = 59049$. For a large number of inputs (in practice, $n > 2$) the parameters of fuzzy sets cannot be acquired from a system expert and must be tuned with selflearning methods, e.g. in the form of neuro-fuzzy models using measurement information from the system. But, according to (Bossley *et al.*, 1995), self-learning neuro-fuzzy models become impractical when $n > 4$. The parameter tuning of the model comes to be difficult or non-realizable. This situation is caused by a too large number of parameters to be tuned. This was observed by Bellman (1961), the inventor of Dynamic Programming, and called the *curse of dimensionality.* The model with a too large number of tuned parameters is said to be *overparametrized.* Many scientists have been looking since that time for methods which would allow us to overcome this problem.

The first method consists in using a nonlattice partitioning of the input space as k-d trees and quad trees (Bossley, 1995). The second method relies on using local models with different partition lattices in different regions of the input space (Babuška and Verbruggen, 1995; Bossley *et al.*, 1995). Both the methods use rectangular sectors in the input space partition. In (Su *et al.*, 1995; Kwon and Zervakis, 1994) a nonregular, nonrectangular partition with multidimensional RBF-functions is proposed. Many papers propose a fight direction with the curse of dimensionality which consists in simplification of fuzzy models by reduction of less "important" or redundant rules and fuzzy sets (Fukumoto *et al.*, 1995; Ishibuchi *et al.*, 1995; Babuška *et al.*, 1996). These methods base on a regular lattice input space partition and offer a limited reduction of the model complication level.

The author of this paper has followed the way proposed in (Su *et al.*, 1995; Kwon and Zervakis, 1994) for the nonregular, nonrectangular input space partition. This approach is very difficult but it gives the rosiest prospects to surmount the curse of dimensionality. Figure 2(a) shows a system surface with two mountains and two flat regions.

Accuratel "mountains" modelling requires using a larger number of fuzzy sets defining the variables x_1 and x_2, Fig. 2(b). This results in a higher density of the partition lattice in their regions. However, a similar lattice density is also in the flat "lowlands" regions where a smaller one would be sufficient. Each node P_{ij} of the partition lattice in the Mamdani model means one rule and three parameters which have to be tuned. This requires a large number of measurements from the modelled system. Tuning models with large numbers of free parameters is very difficult. Application of nonregular, nonrectangular partition lattices as that of Fig. 2(c) decreases considerably the number of nodes, and thereby the amount of the information about the system needed to determine its parameters. In the regions with a quicker variation of the system surface the nodes can be placed densely and in flat regions sparsely. In the case of triangle sectors the surface above them can be uniquely defined on the basis of three nodes only if it is a linear one, goes through them and its borders are straight line segments. If the rectangular sectors of the input space are replaced by triangular ones, for $n = 2$ the rules of the following type are achieved:

$$\text{IF } \left(x_2 \geq a_0^{ij} + a_1^{ij}x_1\right) \text{ AND } \left(x_2 \geq a_0^{ik} + a_1^{ik}x_1\right) \text{ AND } \left(x_2 \geq a_0^{jk} + a_1^{jk}x_1\right)$$

$$\text{THEN } \left(y = a_0^{ijk} + a_1^{ijk}x_1 + a_2^{ijk}x_2\right) \qquad (4)$$

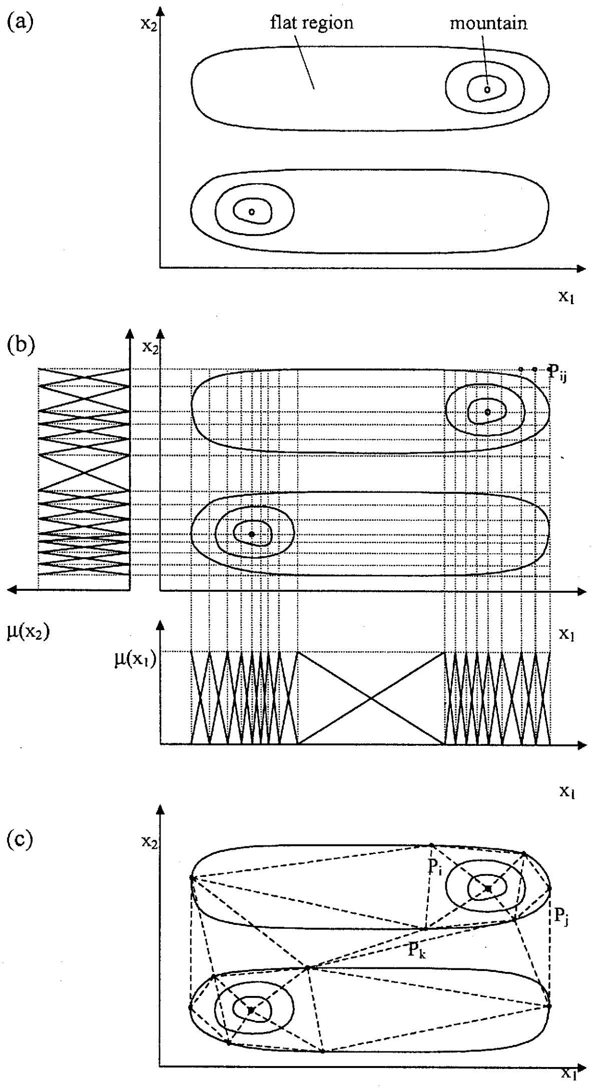


Fig. 2. Contour lines of the surface of a modelled system (a), a rectangular input space partition (b), and a nonregular, nonrectangular one (c).

The rule (4) concerns the sector stretched on three nodes P_i, P_j, P_k (Fig. 2(c)). In the case of the space of a higher dimensionality ($n > 2$), the triangular sector is replaced by a simplex being the simplest form of a convex polyhedron which can exist in this space (for $n = 3$ it becomes a tetrahedron), (Dziubiński and Świątkowski, 1980). The inner space of the simplex is defined by the linear forms

$$x_n \geq b_0 + b_1 x_1 + \cdots + b_{n-1} x_{n-1} \tag{5}$$

and the model surface above the simplex by

$$y = c_0 + c_1x_1 + \cdots + c_nx_n \tag{6}$$

Defining the surface (6) above the simplex (5) requires a smaller amount of information ($n+1$ measurements vectors $x_1, \ldots, x_n, y$) than in the case of a rectangular partition where 2^n vectors are required. The increase in the information needed for the model realization is linear for simplexes and exponential for rectangular partitions. Modelling based on simplex rules (4) with linear conclusions is efficient because the rule thickening is made here locally and only in the regions of the greatest curvature of the system surface. A further decrease in the number of rules is conditioned by introducing nonlinear conclusions. This problem is not trivial because simplex spaces are nonrectangular and the borders of the sector surfaces of the neighbouring simplexes must agree (no abrupt break-down of the surface is allowed). Nonlinear conclusions can be introduced owing to simplex delinearization functions (SDL-functions). This concept is presented in Section 3 and advantages of its application are indicated in Section 5.

2. Modelling with the Attachment-Point Method

The modelled MISO system realizes the input/output mapping f_s corresponding to some hypersurface in the input/output space:

$$f_s : \boldsymbol{X} \rightarrow Y, \quad \boldsymbol{X} = X_1 \times \cdots \times X_n \in \mathbb{R}^n, \quad Y \in \mathbb{R}^1 \tag{7}$$

The model of the system realizes the mapping

$$f_m : \boldsymbol{X} \rightarrow Y_m, \quad \boldsymbol{X} = X_1 \times \cdots \times X_n \in \mathbb{R}^n, \quad Y_m \in \mathbb{R}^1 \tag{8}$$

Adaptive modelling can be formulated as an adaptation process of the model surface f_m to the system surface f_s. Let us imagine the system surface f_s as a rigid spherical cap and the model surface f_m as a piece of a flabby soft cloth which should be attached to the cap so that both the surfaces be sufficiently close to one another and the number of attachment points be possibly small. The model cloth can lie at the beginning under the rigid system cap as is shown in Fig. 3. The model surface can be subsequently attached to the system surface at successive points (Fig. 4) which yields a higher and higher model accuracy.

This heuristic method converges and gives good suboptimal solutions overcoming local minima, which was confirmed experimentally by the author and by Ullrich (1997). The ordinates of particular attachment points (AP's) can be optimised with the use of the LS or Gauss-Seidel methods, which increases considerably the model accuracy when the number of AP's is small. At a greater number of AP's this optimisation has moderate influence. The simplex modelling in a two-dimensional input space is known in the literature as the *Delaunay nets* and

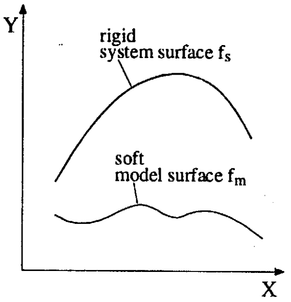


Fig. 3. Initial state of the modelling.

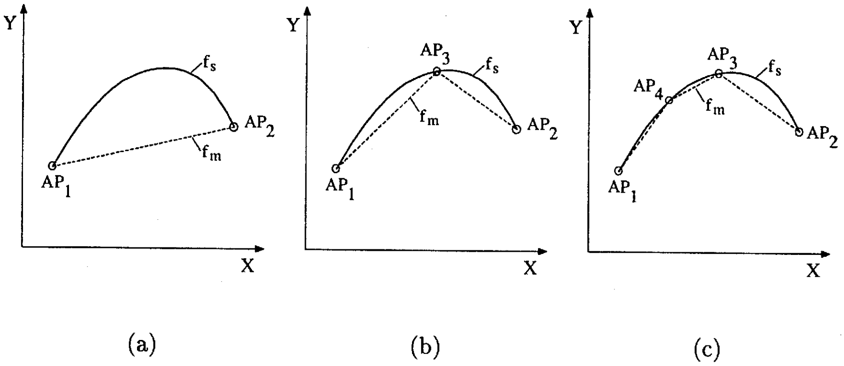
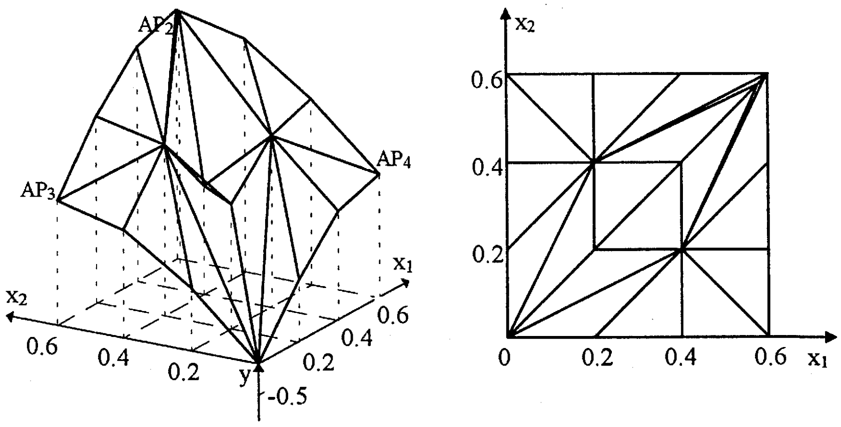


Fig. 4. Increasing the model accuracy in successive steps (f_s—system surface, f_m—model surface).

creating simplex sectors on the basis of AP's as *triangulation.* The AP's being corners of triangular sectors are called the *interpolation nodes* (Ullrich, 1997). The Delaunay nets realize in fact a piecewise linearization. But the "pieces", i.e. simplexes, are found in the input space in a well-defined manner. Their size and placement determines the model structure. A Delaunay net is a self-structuring modelling method. An example of the model surface achieved by triangulation of the input space is shown in Fig. 5.

Each simplex of the input space corresponds to a rule of the type (4) defining in its conclusion the surface above the simplex. Such a model will be efficient if the number of simplexes, which is equal to the number of rules, will be small with a sufficient model accuracy. Only then such models can be used for modelling systems with large numbers of inputs. Ullrich's investigations (Ullrich, 1997) showed that the number of AP's and simplexes generated in Delaunay nets is too high and a modelling system with the number of inputs which is greater than four involves a heavy computational burden. A way to achieve a high accuracy for a small number of simplexes is the application of non-linear interpolation between the AP's. This interpolation can be carried out with simplex delinearization functions (SDL-functions).

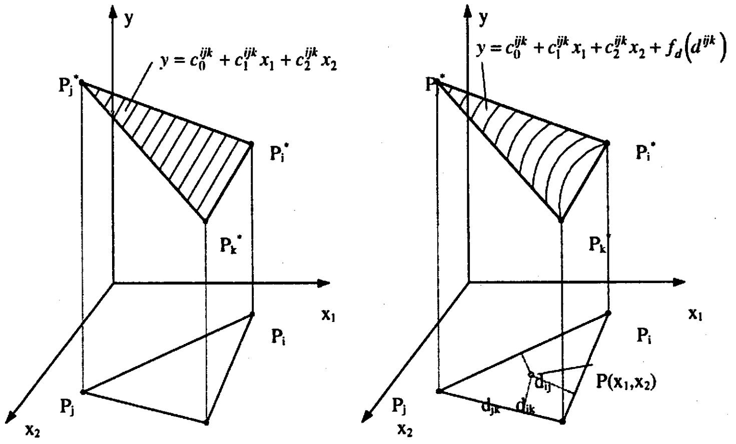


(a) (b)

Fig. 5. Model surface (a) in the space $X_1 \times X_2 \times Y$ achieved by triangularization of the input space $X_1 \times X_2$ and partition of this space by triangular simplexes (b).

3. Simplex Delinearization (SDL) Functions

The idea of the SDL-function is shown in Fig. 6.



(a) (b)

Fig. 6. Simplex interpolation with a linear (a) and a non-linear (b) SDL-function, $P_l = (x_1^l, x_2^l)$, $P_l^* = (x_1^l, x_2^l, y^l)$, $l = i, j, k$.

The simplex linear function (SL-function) shown in Fig. 6(a) is stretched on the nodes P_i^*, P_j^*, P_k^* and satisfies the condition

$$y\left(x_1^l, x_2^l\right) = y_l, \quad l = i, j, k \tag{9}$$

Its borders are line segments at which one SL-function adjoins the neighbouring ones. This assures the continuity of the global model surface. Delinearization of the SL-function

$$y = c_0^{ijk} + c_1^{ijk} x_1 + c_2^{ijk} x_2 \tag{10}$$

consists in adding a non-linear function f_d which is non-zero above the simplex, but equals zero on its triangular borders. Then the borders are still line segments. The formulae

$$f_d = c_3 d^{ijk}, \quad f_d = c_3 d^{ijk} + c_4\left(d^{ijk}\right)^2, \quad f_d = c_3 d^{ijk} + c_4\left(d^{ijk}\right)^2 + c_5\left(d^{ijk}\right)^3 \tag{11}$$

constitute three variants of the SDL-function with different numbers of degrees of freedom. Clearly, other forms of the SDL-function are also possible. Here d^{ijk} is the product of the distances of the point $P(x_1, x_2)$ lying inside the simplex from its borders (Fig. 6(b)).

$$d^{ijk} = d^{ij} d^{ik} d^{jk} \tag{12}$$

If a simplex border passes through its nodes P_i, P_j and is described by the general line equation

$$A^{ij} x_1 + B^{ij} x_2 + C^{ij} = 0 \tag{13}$$

then this equation can be transformed into normal form

$$x_1 \cos\left(\alpha^{ij}\right) + x_2 \sin\left(\alpha^{ij}\right) - p^{ij} = 0 \tag{14}$$

where

$$\cos\left(\alpha^{ij}\right) = \frac{A}{\sqrt{A^2 + B^2}}, \quad \sin\left(\alpha^{ij}\right) = \frac{B}{\sqrt{A^2 + B^2}}, \quad p^{ij} = \frac{C}{\sqrt{A^2 + B^2}}$$

The distance d^{ij} of a point $P(x_1, x_2)$ lying inside a simplex can be calculated according to the formula

$$d^{ij} = \left|x_1 \cos\left(\alpha^{ij}\right) + x_2 \sin\left(\alpha^{ij}\right) - p^{ij}\right| \tag{15}$$

Adding the SL-function (10) to the SDL-function (11), we get the simplex non-linear (SNL) function

$$y = c_0^{ijk} + c_1^{ijk} x_1 + c_2^{ijk} x_2 + f_d\left(d^{ijk}\right) \tag{16}$$

The parameters $c_0^{ijk}, c_1^{ijk}, c_2^{ijk}$ are determined by the corner coordinates of the simplex $P_l^* = \left(x_1^l, x_2^l, y^l\right)$, $l = i, j, k$ on which the plane (10) is stretched. The parameters $c_3, c_4, \ldots$ of the SDL-function are determined via the least-squares (LS) method using a pseudo-inverse matrix $\boldsymbol{D}$. If s measurement points $P_m^* = (x_{1m}, x_{2m}, y_m)$,

$m = 1, \ldots, s$ (apart from the AP's), are situated in the simplex, then for each of them the difference between the measured output value y_m and y, from the linear interpolation (10), can be calculated as

$$\Delta_m = y_m - \left(c_0^{ijk} + c_1^{ijk} x_{1m} + c_2^{ijk} x_{2m}\right) \tag{17}$$

and the vector Δ of the function differences

$$\Delta = \begin{bmatrix} \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_s \end{bmatrix} \tag{18}$$

can be created.

For each measurement point P_m its distance product d_m can be calculated from (12) and, accordingly, the matrix

$$\boldsymbol{D} = \begin{bmatrix} d_1 & d_1^2 & \cdots & d_1^r \\ d_2 & d_2^2 & \cdots & d_2^r \\ \vdots & \vdots & & \vdots \\ d_s & d_s^2 & \cdots & d_s^r \end{bmatrix} \tag{19}$$

can be created.

If $\boldsymbol{C}$ signifies the vector of the unknown parameters of the SDL-function

$$\boldsymbol{C} = \begin{bmatrix} c_3 \\ c_4 \\ \vdots \\ c_{3+r} \end{bmatrix} \tag{20}$$

then its optimal estimate can be calculated from

$$\boldsymbol{C}_{\text{opt}} = \left(\boldsymbol{D}^T \boldsymbol{D}\right)^{-1} \boldsymbol{D}^T \Delta = \boldsymbol{D}^+ \Delta \tag{21}$$

4. Algorithm of the Simplex Modelling with AP's and SDL-Functions for Two-Input Systems

Now, we are ready to present the following algorithm:

1. **Pre-processing the input/output measurement data of the modelled system.** This is to get a unique system representation and to decrease the amount of data.

2. **Determining a basic model $y^* = M_I(x_1, x_2)$.**

2.1. Determining the measurement points P_j with minimal and maximal values of the system input x_1:

$$\begin{cases} P_1 = \begin{bmatrix} x_{11} \\ x_{21} \\ y_1 \end{bmatrix}, \quad x_{11} \le x_{1j} \\ P_2 = \begin{bmatrix} x_{12} \\ x_{22} \\ y_2 \end{bmatrix}, \quad x_{12} \ge x_{1j}, \quad j = 1, \dots, m \end{cases} \tag{22}$$

where m stands for the number of the measurement points representing the system surface after pre-processing. If the number of such points is greater than two, the pair P_1, P_2 with the greatest Euclidean distance R_{12} in the input space should be found (Fig. 7),

$$R_{12} = \left(\sum_{i=1}^{2} \left(x_{i1} - x_{i2} \right)^2 \right)^{1/2} \tag{23}$$

The points P_1 and P_2 become the first AP's: $P_1 \to AP_1$, $P_2 \to AP_2$.

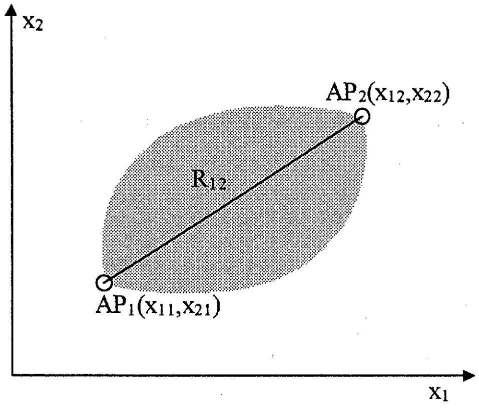


Fig. 7. The first AP's of the basic model.

2.2. Determining the point P_3 for which the sum of the distances

$$R_{312} = R_{31} + R_{32} = \left(\sum_{i=1}^{2} \left(x_{i3} - x_{i1} \right)^2 \right)^{1/2} + \left(\sum_{i=1}^{2} \left(x_{i3} - x_{i2} \right)^2 \right)^{1/2} \tag{24}$$

from the points AP_1 and AP_2 in the input space is the greatest one (Fig. 8). After finding $P_3(x_{13}, x_{23}, y_3)$, we should check through (24) if it lies on the line connecting AP_1 and AP_2. If the condition

$$\frac{x_{23} - x_{21}}{x_{13} - x_{11}} = \frac{x_{22} - x_{21}}{x_{12} - x_{11}} \tag{25}$$

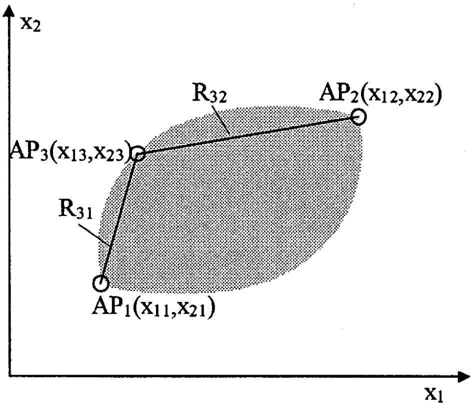


Fig. 8. Step 2.2 of constructing the basic model.

is satisfied, all the measurement points lie on one line in the input space and the modelling can be reduced, by rotation of the coordinates, to a two-dimensional problem. Otherwise, the fourth AP is to be determined.

2.3. Determining the point P_4 for which the sum R_{4123} of the distances from the points AP_1, AP_2, AP_3 in the input space is the greatest one (Fig. 9).

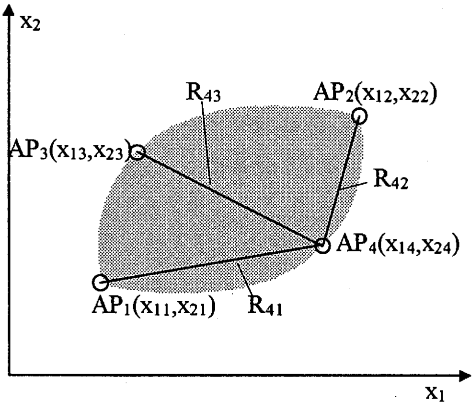


Fig. 9. Step 2.3 of constructing the basic model.

After finding P_4, a convexity condition for the quadrangle (AP_1, AP_2, AP_3, AP_4) must be checked. This can be achieved by checking if the points AP_3 and P_4 lie on the same side of the line connecting AP_1 and AP_2. One of the conditions

$$\left(\frac{x_{23}-x_{21}}{x_{13}-x_{11}} > \frac{x_{22}-x_{21}}{x_{12}-x_{11}}\right) \text{ AND } \left(\frac{x_{24}-x_{21}}{x_{14}-x_{11}} < \frac{x_{22}-x_{21}}{x_{12}-x_{11}}\right) \tag{26}$$

and

$$\left(\frac{x_{23}-x_{21}}{x_{13}-x_{11}} < \frac{x_{22}-x_{21}}{x_{12}-x_{11}}\right) \text{ AND } \left(\frac{x_{24}-x_{21}}{x_{14}-x_{11}} > \frac{x_{22}-x_{21}}{x_{12}-x_{11}}\right) \tag{27}$$

must be satisfied.

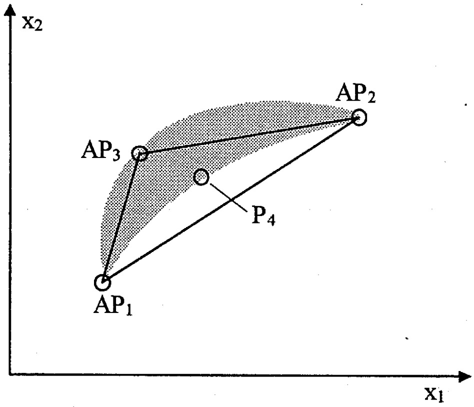


Fig. 10. A basic model with three AP's.

If these conditions are not satisfied, the point P_4 lies inside the triangle (AP_1, AP_2, AP_3), Fig. 10, and cannot be accepted as a new AP. The basic model will then be defined by three AP's. Go to Step 4. If the point P_4 satisfies (26) or (27), it is accepted as the new AP: $P_4 \rightarrow AP_4$, Fig. 11.

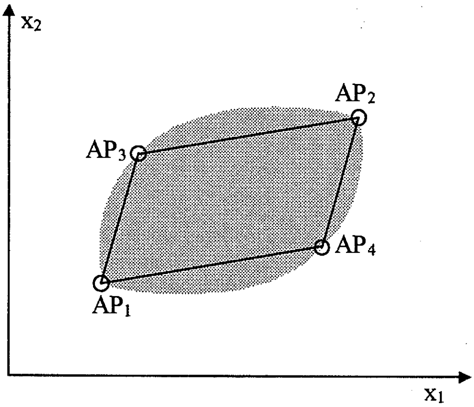


Fig. 11. A basic model with four AP's.

3. **Partition of the input space into sectors.** The input space inside the quadrangle $(AP_1, \ldots, AP_4)$, Fig. 12, is partitioned into two triangular sectors

I_1 and I_2. The partition can be made in two ways. According to Delaunay's net theory, the shorter diagonal is chosen. The outer space of the basic model is partitioned by the lines coming out of the intersection point P_R of the diagonals and going through the outer AP's, Fig. 13. The space lying in the outer sectors O_i will be related to the model of the nearest neighbouring simplex. In the case of the basic model, the models of the sectors O_1 and O_2 are identical. After splitting the basic model into a greater number of inner sectors, the situation can be different.

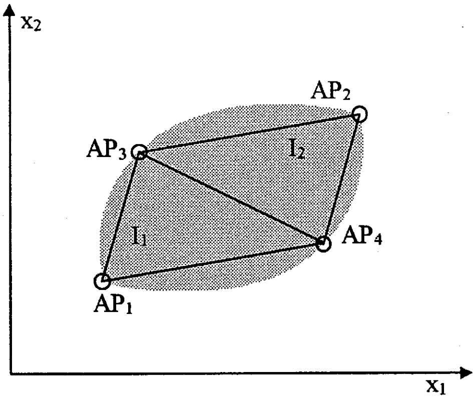


Fig. 12. Partition of the input space of a basic model into inner sectors I_i.

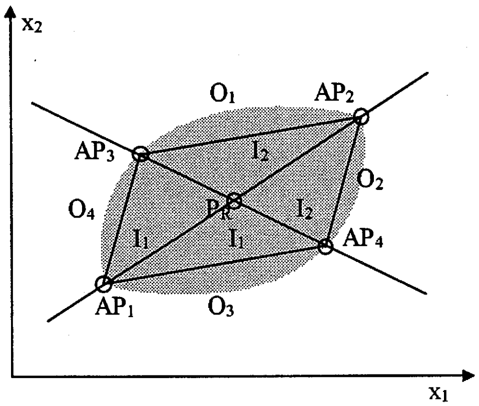


Fig. 13. Partition of the outer space of the model into outer sectors O_i.

4. **Determining local models.** From the partition of the input space, the basic model M_I is determined:

$$M_I = M_{I1} + M_{I2} \tag{28}$$

The model M_{I1} refers to the sector I_1 and to adjacent outer sectors O_3 and O_4:

$$M_{I1} = M_{O3} = M_{O4} \tag{29}$$

Similarly, the model M_{I2} refers to the sector I_2 and the outer regions O_1 and O_2:

$$M_{I2} = M_{O1} = M_{O2} \tag{30}$$

Both the models have the form of the rules

$$\begin{aligned}
&M_{I1}(AP_1, AP_3, AP_4,):\\
&\quad \text{IF } \left(x_2 R_{13}(d_0^{13} + d_1^{13} x_1)\right) \text{ AND } \left(x_2 R_{14}(d_0^{14} + d_1^{14} x_1)\right) \\
&\quad \text{AND } \left(x_2 R_{34}(d_0^{34} + d_1^{34} x_1)\right) \text{ THEN } \left(y = D_0^{134} + D_1^{134} x_1 + D_2^{134} x_2\right)
\end{aligned} \tag{31}$$

and

$$\begin{aligned}
&M_{I2}(AP_2, AP_3, AP_4,):\\
&\quad \text{IF } \left(x_2 R_{23}(d_0^{23} + d_1^{23} x_1)\right) \text{ AND } \left(x_2 R_{24}(d_0^{24} + d_1^{24} x_1)\right) \\
&\quad \text{AND } \left(x_2 R_{34}(d_0^{34} + d_1^{34} x_1)\right) \text{ THEN } \left(y = D_0^{234} + D_1^{234} x_1 + D_2^{234} x_2\right)
\end{aligned} \tag{32}$$

which define surfaces over each simplex of the input space. Here R_{ij} stands for the relation $\geq$ or $\leq$ determined for the points $P_m(x_1, x_2)$ lying inside the simplex. Particular component premises of the type $(x_2 R_{ij}(d_0^{ij} + d_1^{ij} x_1))$ mean the right or left sides of the lines going through AP_i and AP_j. The direction of the relations ($\geq$, $\leq$) can be found by inserting the coordinates of the third simplex corner AP_k in the input space $X_1 \times X_2$ into

$$x_2 = d_0^{ij} + d_1^{ij} x_1 \tag{33}$$

which represents the line going through the points AP_i and AP_j. The coefficients $D_0^{ijk}, D_1^{ijk}, D_2^{ijk}$ in the rules' conclusions (31), (32) are the coefficients of the plane equation going through the points AP_i, AP_j, AP_k in the space $X_1 \times X_2 \times Y$.

5. **Accuracy testing of the global model.** For each input measurement vector $\boldsymbol{W}_j^T = [x_{1j}, x_{2j}]$, $j = 1, \dots, m$, a model output $y^*(x_{1j}, x_{2j})$ is calculated on the basis of only one local model (the input vector can be situated on the border of two simplexes) corresponding to the input-space simplex in which lies the vector $\boldsymbol{W}_j$. Then the model error E_j of the output y^* is determined according to

$$E_j = y(x_{1j}, x_{2j}) - y^*(x_{1j}, x_{2j}) \tag{34}$$

where y is the measured system output. The absolute value of the mean error $|E|_{\text{mean}}$ for all the model outputs y_j^* is calculated from

$$|E|_{\text{mean}} = \frac{1}{m}\sum_{j=1}^{m} |E_j| \tag{35}$$

If the mean model error is sufficiently small, i.e.

$$|E|_{\text{mean}} \le |E|_{\text{ad}} \tag{36}$$

where $|E|_{\text{ad}}$ denotes an admissible model error, then the modelling is finished. Otherwise, go to Steps 6 or 7.

6. **Delinearization of simplexes with insufficient modelling accuracy.** The delinearization is carried out according to instructions given in Section 3. It can be accomplished after any modelling cycle. The decision belongs to us. Go to Step 5.

7. **Determining new AP's.** A measurement point $P_p(x_{1p}, x_{2p}, y_p)$ should be found for which the model output error E_p is the greatest and positive one, and a point $P_n(x_{1n}, x_{2n}, y_n)$ for which the error is the smallest and negative one, i.e.

$$E_p(x_{1p}, x_{2p}, y_p) = \max E_j(x_{1j}, x_{2j}, y_j), \quad E_p > 0, \quad j = 1, \dots, m \tag{37}$$

and

$$E_n(x_{1n}, x_{2n}, y_n) = \min E_j(x_{1j}, x_{2j}, y_j), \quad E_n < 0, \quad j = 1, \dots, m \tag{38}$$

The points P_p and P_n become new AP's ($P_p \to AP_p$, $P_n \to AP_n$) and we have to change the corresponding local models. If a few points have the same maximal (minimal) error value, then only one of them is chosen as an AP. Go to Step 8.

8. **Decomposition of local models with new AP's.** If the new AP belongs to an inner simplex, then it should be decomposed into three smaller simplexes as shown in Fig. 14.

The previous local model M_i, where i is the number of the local model, is decomposed into three new models $M_{(i)}$:

$$\begin{aligned} M_i(AP_i, AP_j, AP_k)M(i) \to (AP_i, AP_j, AP_p) &+ M_{(i+1)}(AP_i, AP_p, AP_k) \\ &+ M_{(i+2)}(AP_j, AP_k, AP_p) \end{aligned} \tag{39}$$

The new local models can be constructed as described in Step 4. If a new AP lies in the outer sector O_r, a new simplex is constructed on the basis of the new AP and two outer neighbouring AP's, Fig. 15.

A new local model for the new simplex (AP_2, AP_3, AP_p) is constructed according to Step 4. Go to Step 5.

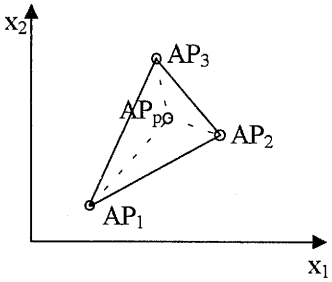


Fig. 14. Decomposition of an inner simplex of the input space after introducing a new AP (AP_p).

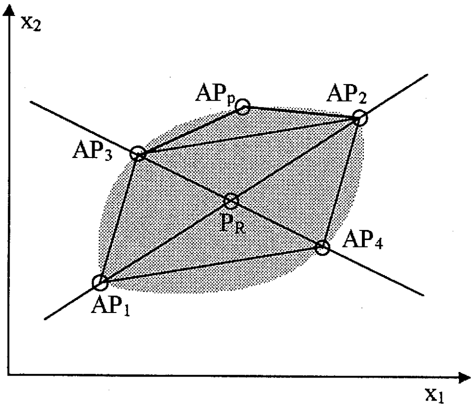


Fig. 15. Creating a new outer simplex.

5. Experiment

In the first experiment, the modelling with attachment points without delinearizing SDL-functions will be shown. In the second experiment, the SDL function will be applied for the accuracy improvement of the model without increasing the number of rules. The measurement information about the modelled system is given in the form of the measurement vectors in Tab. 1.

Table 2 shows the modelling, successively determined AP's and a full AP set in each modeling cycle. As Tab. 2 and Fig. 16 show, increasing the number of AP's results in an increased model accuracy.

In the case of a practical application, we can choose a model which assures the required accuracy for a possibly small number of AP's. The basic model with four AP's and the error $|E|_{\text{mean}} = 0.184$ is shown in Fig. 17. The global model in this

Table 1. Input/output measurements of the modelled MISO-system.

P_i	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}	P_{16}	P_{17}
x_1	0	0.2	0.4	0.6	0	0.2	0.4	0.6	0	0.2	0.4	0.6	0	0.2	0.4	0.577	0.6
x_2	0	0	0	0	0.2	0.2	0.2	0.2	0.4	0.4	0.4	0.4	0.6	0.6	0.6	0.577	0.6
y	0	0.499	0.873	0.998	0.499	0.998	1.372	1.496	0.873	1.372	0.873	1.876	0.998	1.496	1.876	2	1.955

modelling cycle consists of two local models, $M_I = M_{I1} + M_{I2}$, with rules

$M_{I1}(AP_1, AP_2, AP_3)$:

$$\text{IF } (x_1 \leq 0) \text{ AND } (x_1 \leq x_2) \text{ AND } (x_2 \leq 0.6) \text{ THEN } (y = 1.662x_1 + 1.662x_2) \quad (40)$$

and

$M_{I2}(AP_1, AP_2, AP_4)$:

$$\text{IF } (x_1 \leq 0.6) \text{ AND } (x_1 \geq x_2) \text{ AND } (x_1 \geq 0) \text{ THEN } (y = 1.662x_1 + 1.662x_2) \quad (41)$$

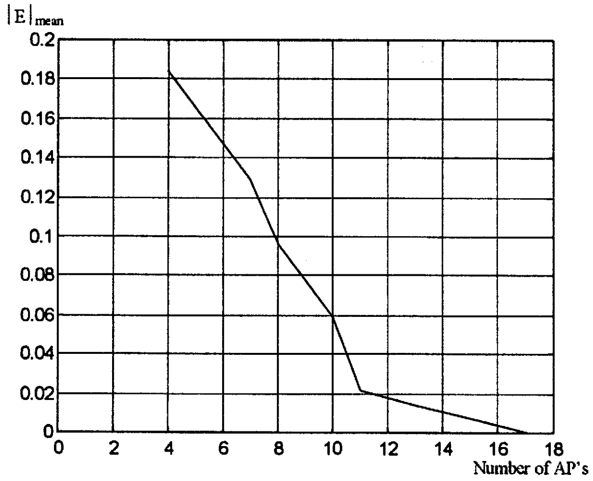


Fig. 16. Model accuracy versus the number of attachment points.

Table 2. Modelling cycles of the MISO-system (2 inputs) with the AP-method and without SDL-functions.

AP's		AP_1	AP_{14}	AP_{11}	AP_4	AP_{15}	AP_8	AP_5	AP_{16}	AP_{12}	AP_6	AP_7	AP_9	AP_3	AP_{17}	AP_{10}	AP_{13}	AP_2	
AP's	x_1	0	0.2	0.4	0.6	0	0.2	0.4	0.6	0	0.2	0.4	0.6	0	0.2	0.4	0.577	0.6	$\lvert E\rvert_{\text{mean}}$
in the	x_2	0	0	0	0	0.2	0.2	0.2	0.2	0.4	0.4	0.4	0.4	0.6	0.6	0.6	0.577	0.6	
model	y	0	0.499	0.873	0.998	0.499	0.998	1.372	1.496	0.873	1.372	0.873	1.876	0.998	1.496	1.876	2	1.955	0
AP_1··	y_I^*	0	0.332	0.665	0.998	0.332	0.665	0.997	1.330	0.665	0.997	1.333	1.662	0.998	1.330	1.662	1.919	1.955	
··AP_4	E_I	0	0.167	0.208	0	0.167	0.333	0.375	0.166	0.208	0.375	-0.46	0.214	0	0.166	0.214	0.081	0	0.184
AP_1··	y_{II}^*	0	0.332	0.665	0.998	0.332	0.436	1.372	1.330	0.665	1.372	0.873	1.662	0.998	1.330	1.662	1.870	1.955	
··AP_7	E$_{II}$	0	0.167	0.208	0	0.167	0.561	0	0.166	0.208	0	0	0.214	0	0.166	0.214	0.130	0	0.129
AP_1··	Y$_{III}^*$	0	0.332	0.665	0.998	0.332	0.998	1.372	1.330	0.665	1.372	0.873	1.662	0.998	1.330	1.662	1.870	1.955	
··AP_8	E$_{III}$	0	0.167	0.208	0	0.167	0	0	0.166	0.208	0	0	0.214	0	0.166	0.214	0.130	0	0.096
AP_1··	y_{IV}^*	0	0.332	0.665	0.998	0.332	0.998	1.372	1.437	0.665	1.372	0.873	1.876	0.998	1.437	1.876	1.870	1.955	
·AP_{10}	E$_{IV}$	0	0.167	0.208	0	0.167	0	0	0.059	0.208	0	0	0	0	0.059	0	0.130	0	0.059
AP_1··	y_V^*	0	0.436	0.873	0.998	0.436	0.998	1.372	1.437	0.873	1.372	0.873	1.876	0.998	1.437	1.876	1.870	1.955	
·AP_{12}	E$_V$	0	0.063	0	0	0.063	0	0	0.059	0	0	0	0	0	0.059	0	0.130	0	0.022
AP_1··	y_{VI}^*	0	0.436	0.873	0.998	0.436	0.998	1.372	1.437	0.873	1.372	0.873	1.876	0.998	1.437	1.876	2	1.995	
·AP_{13}	E$_{VI}$	0	0.063	0	0	0.063	0	0	0.059	0	0	0	0	0	0.059	0	0	0	0.014
AP_1··	y_{VII}^*	0	0.499	0.873	0.998	0.499	0.998	1.372	1.437	0.873	1.372	0.873	1.876	0.998	1.437	1.876	2	1.955	
·AP_{15}	E$_{VII}$	0	0	0	0	0	0	0	0.059	0	0	0	0	0	0.059	0	0	0	0.007
AP_1··	y_{VIII}^*	0	0.499	0.873	0.998	0.499	0.998	1.372	1.496	0.873	1.372	0.873	1.876	0.998	1.496	1.876	2	1.955	
·AP_{17}	E$_{VIII}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

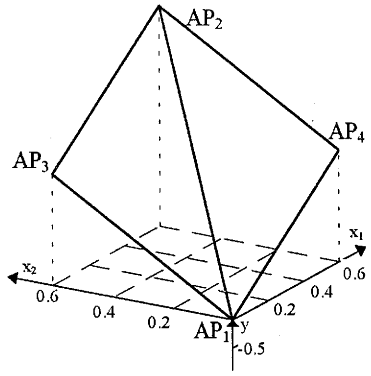


Fig. 17. Surface of the basic model with four AP's, $|E|_{\text{mean}} = 0.184$.

Figure 18 shows the model surface with eight AP's achieved in the third cycle of modelling, with $|E|_{\text{mean}} = 0.096$. This model consists of ten simplexes and ten rules of the type (40). In Fig. 19, the model surface achieved in the last modelling step, with seventeen AP's and twenty simplexes (rules), is shown. In this case, we have $|E|_{\text{mean}} = 0$.

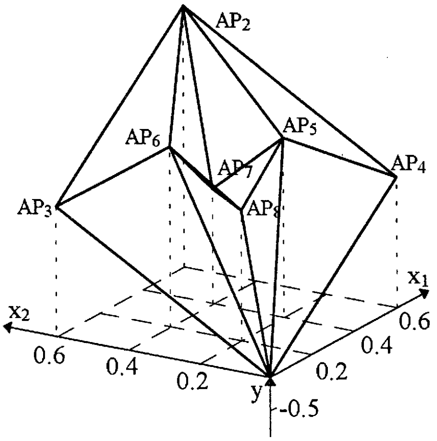


Fig. 18. Model surface for eight AP's, $|E|_{\text{mean}} = 0.096$.

To increase the modelling accuracy, we can additionally apply SDL-functions after any modelling cycle with the AP-method. Figure 20 shows the input-space partition for which the mean absolute error is equal to 0.076.

After delinearization of the functions f for the same input-space partition as in Fig. 20, the zero value of the mean error is achieved. The resulting simplex non-linear

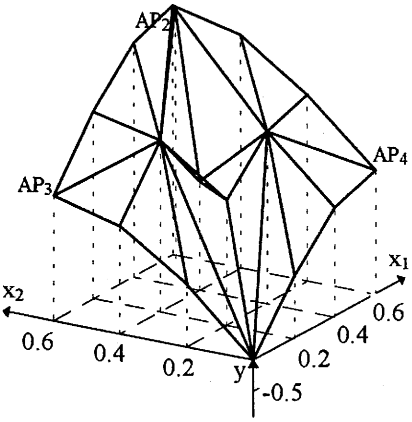


Fig. 19. Model surface with 20 simplexes (rules), $|E|_{\text{mean}} = 0$.

(SNL) functions including SDL-ones are given by

$$
\begin{aligned}
f1 &: \ y = 0.623 + 2.495\,x_1 + 0.625\,x_2 \\
f2 &: \ y = 0.626 + 2.49\,x_1 + 0.62\,x_2 \\
f3 &: \ y = 0.8945 + 1.1475\,x_1 + 0.62\,x_2 + 8.417\,d_{26}\,d_{69} \\
f4 &: \ y = 0.789 + 0.9717(x_1 + x_2) \\
&\qquad +2.269 \times 10^3 d_{26} d_{25} d_{56} - 2.5506 \times 10^6 (d_{26}\,d_{25}\,d_{56})^2 \\
f5 &: \ y = 0.8945 + 0.62\,x_1 + 1.1475\,x_2 + 8.417\,d_{25}\,d_{58} \\
f6 &: \ y = 2.495\,x_1 + 2.1825\,x_2 + 3.4955\,d_{16}\,d_{6,10} \\
f7 &: \ y = 2.2867(x_1 + x_2) + 73.727\,d_{15}\,d_{16}\,d_{56} \\
f8 &: \ y = 2.2825\,x_1 + 2.495\,x_2 + 3.4955\,d_{15}\,d_{57} \\
f9 &: \ y = 0.623 + 0.625\,x_1 + 2.495\,x_2 \\
f10 &: \ y = 0.623 + 0.62\,x_1 + 2.49\,x_2
\end{aligned}
\tag{42}
$$

where particular SDL-functions are given by

$$
\begin{aligned}
d_{15} &= \left|(-0.5x_1 + x_2)/1.25^{0.5}\right| \\
d_{16} &= \left|(2x_1 - x_2)/5^{0.5}\right| \\
d_{36} &= d_{56} = d_{46} = \left|(x_1 + x_2 - 0.6)/2^{0.5}\right| \\
d_{26} &= \left|(0.5x_1 - x_2 + 0.3)/1.25^{0.5}\right| \\
d_{58} &= |x_2 - 0.2| \\
d_{57} &= |x_1 - 0.4| \\
d_{69} &= |x_1 - 0.2| \\
d_{6,10} &= |x_2 - 0.4|
\end{aligned}
\tag{43}
$$

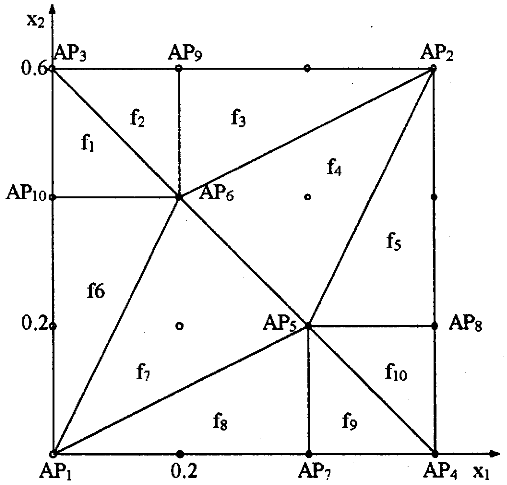


Fig. 20. Partition of the input space into simplexes, $|E|_{\text{mean}} = 0.076$.

It should be noticed that the model with ten rules including conclusions (42) has the same accuracy as the model with linear conclusions and twenty rules. The surfaces of both the models are shown in Fig. 21.

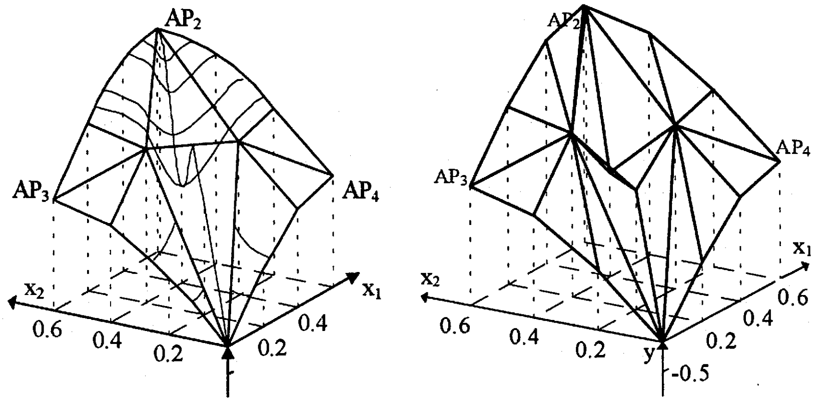


Fig. 21. Comparison of the simplex model with SDL-functions (a) and with linear functions in the rules conclusions (b), $|E|_{\text{mean}} = 0$.

6. Conclusions

Rectangular partitions of the input space result in models with large numbers of rules. A manner to decrease the number of rules is partition of the space into nonregular simplexes with linear functions defining the model surface, which is a method applied in Delaunay's nets. Further decreasing the number of rules and simplexes is possible with SDL-functions described in the paper. Their advantageous influence is confirmed by experiments.

References

Babuška R., Verbruggen H.B. (1995): *Identification of composite linear models via fuzzy clustering.* — Proc. 3rd *European Control Conference*, Rome, Italy, pp.1207–1212.

Babuška R., Setnes M., Kaymak U., van Nauta Lemke H.R. (1996): *Simplifications of fuzzy rule bases.* — Proc. *EUFIT'96*, Aachen, Germany, pp.1115–1119.

Bellman R.E. (1961): *Adaptive Control Processes.* — Princetown: Princetown University Press.

Bossley K.M., Brown M., Harris C.J. (1995): *Neurofuzzy model construction for the modelling of non-linear processes.* — Proc. 3rd *European Control Conference*, Rome, Italy, pp.2438–2443.

Dziubiński I., Świątkowski T. (1980): Mathematical Handbook. — Warsaw: PWN (in Polish).

Fukumoto S., Miyajima H., Kishida K., Nagasawa Y. (1995): *A destructive learning method of fuzzy inference rules.* — Proc. Int. Conf. *FUZZ-IEEE, IFES'95*, pp.687–694.

Ishibuchi H., Nozaki K., Yamamoto N., Tanaka H. (1995): *Selecting fuzzy if-then rules for classification problems using genetic algorithms.* — IEEE Trans. on Fuzzy Syst., Vol.3, No.3, pp.260–270.

Kwon T.M., Zervakis M.E. (1994): *A self-organizing KNN fuzzy controller and its neural network structure.* — Int. J. Adap. Contr. Sign. Process., Vol.8, No.4, pp.407–431.

Su M.C., Kao C.J., Liu K.M., Liu C.Y. (1995): *Rule extraction using a novel class of fuzzy degraded hyperellipsoidal composite neural networks.* — Proc. Int. Conf. *FUZZ-IEEE, IFES'95*, pp.233–238.

Ullrich T. (1997): *Datengetriebene Modellierung nichtlinearer Strecken mit Delaunay-Netzen.* — Automatisierungstechnik, No.45/5, Oldenburg Verlag, Germany.

Zeng X.J., Singh M.G. (1996): *Decomposition property of fuzzy systems and its applications.* — IEEE Trans. Fuzzy Syst., Vol.4, No.2, pp.149–163.

Received: 26 September 1997
Revised: 2 June 1998