Hierarchical Look-up Tables

Susanne Ernst
Darmstadt University of Technology
Institute of Automatic Control
Laboratory of Control Engineering and Process Automation
Landgraf-Georg-Straße 4, D-64289 Darmstadt
Phone: +49-6151-164524, Fax: +49-6151-293445
email:{SErnst}@iat.tu-darmstadt.de

ABSTRACT: Look-up tables are the most common type of nonlinear static models in practical applications. They consist of a number of data points positioned on a multi-dimensional grid. Due to their easy structuring, look-up tables allow the implementation on low-cost hardware for applications under real-time conditions. However, since the locations of data points are prescribed by the grid structure, memory demands grow exponentially with the input space dimensionality. Therefore, practical applications are limited to the usage of look-up tables with one- or two-dimensional input spaces. In this paper hierarchical look-up tables with a heuristic strategy for data point positioning are proposed. Their data point distribution is adjusted to the local complexity of the nonlinear function which has to be modelled. Consequently, hierarchical look-up tables do not suffer from the curse of dimensionality and can be applied to cope with high-dimensional input spaces.

KEYWORDS: look-up table, linear interpolation, hierarchical structure optimisation, input space decomposition

1 INTRODUCTION

For real-time applications, efficient methods for representation of nonlinear characteristics of process models or control strategies are required. Especially in automotive engineering, increasing demands concerning safety aspects, passenger comfort or exhaust gas emission make use of advanced automotive control systems necessary. Information processing in such control units for engine management, emission control or tracking control for instance is based on the integration of models representing nonlinear relationships.
Artificial neural networks (ANN) can generally be applied to the simulation of nonlinear behaviour. These models possess the capability to learn any nonlinear mappings from measurement data. If efficient parameter and structure optimisation methods are applied to the neural network architectures, only few parameters can describe highly nonlinear behaviour. Despite of this, training of ANN is often difficult and requires expert knowledge to yield high training and generalisation quality. Moreover, for the simulation of ANN complex analytical formulas have to be calculated. These drawbacks restrict applicability of ANN in industry. In the field of automotive applications, due to cost reasons, computational resources are strongly restricted. Therefore, automotive control systems are equipped with relatively simple hardware components. The implementation of nonlinear mappings has to meet the following constraints:

- limited computational resources,
- limited storage capacity,
- real-time operation.

Because of these constraints, neural network models are unfeasible. The state-of-the-art approach for nonlinear static mappings under conditions mentioned above are *look-up tables*. These models have an extremely simple structure given by a set of data points. As a result, the output calculation of look-up tables can be carried out by fast and few operations on low-cost hardware. Automotive control units of modern vehicles contain about 100 such look-up tables for the realisation of nonlinear mappings. Since the data points of look-up tables are located on multi-dimensional grids, these mappings suffer from the *curse of dimensionality* problem. The most obvious manifestation occurs in the exponentially increasing storage requirements for an increasing dimensionality of the input space. This disadvantage prevents the implementation of grid-based look-up tables for modelling problems with more than two input variables. If higher-dimensional mappings have to be realised, then the outputs of a respective number of low-dimensional look-up tables are combined in an appropriate way. However the construction of such high-dimensional mappings requires physical insights. In this paper a novel approach for data point location is proposed. Contrary to the grid-based structure of classical look-up tables, the proposed *hierarchical look-up tables* allow the placement of data points with respect to the local complexity of the underlying nonlinearity. This is the essential premise to avoid the curse of dimensionality.

2 GRID-BASED LOOK-UP TABLES

Look-up tables are defined by a set of data points or nodes. Each (n+1)-dimensional node comprises two components. The scalar *weight* (or height) of a data point is an estimate of the underlying nonlinear function at the data point *position* in the n-dimensional input space. All nodes are located on an axis-orthogonal lattice structure (Figure 1). From this grid follows the partitioning of the input space into hypercubes. The resulting hypercubes are defined by their enclosing 2^n data points which represent the vertices of the grid element.
The data point weights are identical with the look-up table outputs at their corresponding data point positions. For arbitrary input vectors, *interpolation* is performed to yield the single-valued model output. Contrary to neural networks models strictly local information is exploited by using exclusively the enclosing 2^n nodes for the interpolation task. Since a linear hyperplane is given by (n+1) points in the n-dimensional space, only for look-up tables with one input variable a pure linear interpolation can be performed between the active data points. In the most common two-dimensional case, bilinear interpolation is performed with the equation (Schmitt, 1995):

$$
\begin{aligned}
&\hat{y} = w_{i,j}\Phi_{i,j} + w_{i+1,j}\Phi_{i+1,j} + w_{i,j+1}\Phi_{i,j+1} + w_{i+1,j+1}\Phi_{i+1,j+1} \\
&\textit{with} \quad \Phi_{i,j} = \frac{a_{i+1,j+1}}{a},\ \Phi_{i+1,j} = \frac{a_{i,j+1}}{a},\ \Phi_{i,j+1} = \frac{a_{i+1,j}}{a},\ \Phi_{i+1,j+1} = \frac{a_{i,j}}{a} \\
&\qquad a_{i,j} = (u_1 - u_{1i})(u_2 - u_{2j}), \qquad a_{i+1,j} = (u_{1i+1} - u_1)(u_2 - u_{2j}) \\
&\qquad a_{i,j+1} = (u_1 - u_{1i})(u_{2j+1} - u_2), \qquad a_{i+1,j+1} = (u_{1i+1} - u_1)(u_{2j+1} - u_2) \\
&\qquad a = a_{i,j} + a_{i+1,j} + a_{i,j+1} + a_{i+1,j+1}
\end{aligned}
\tag{2.1}
$$

Here the height $w_{i,j}$ of a data point is weighted with the quotient $\Phi_{i,j}(\mathbf{u})$ of the opposite area $a_{i+1,j+1}(\mathbf{u})$ and the total area a, see Figure 2a. The four terms $\Phi_{i,j}(\mathbf{u})$ of course sum up to one. Formula (2.1) can be easily extended to arbitrary input space dimensions. The resulting hypercube interpolation rules can be rewritten as polynomials of order n where all terms containing at least one input variable of order two or higher are discarded. Due to this, linear interpolation can be found orthogonal to each input dimension. Therefore, output interpolation can also be reconstructed into a recursive way by successive application of linear interpolation to all input space dimensions separately (Figure 2b). Note that, since interpolation is carried out with respect to a finite number of active nodes, piecewise linear interpolation behaviour of the overall model succeeds. The whole output calculation of look-up tables has to be done in a two step procedure:

1. Selection of the enclosing nodes
2. Output interpolation

The first step can be carried out extremely fast, since the regular arrangement of node positions allows access from input space positions to the memory location of the enclosing data points. If an equidistant grid is chosen (Figure 1a), the memory location can be computed directly by simple address calculation. Otherwise, for each input space direction distinctly a one-dimensional search has to be done selecting the data point positions respectively, see Figure 1b. After that, the address of memory can be calculated. Note that, for grid-based look-up tables the storage of data point weights is sufficient while their data point positions need not explicitly to be stored. The second step comprises output evaluation according to equation (2.1) where only computationally less demanding operations are required.

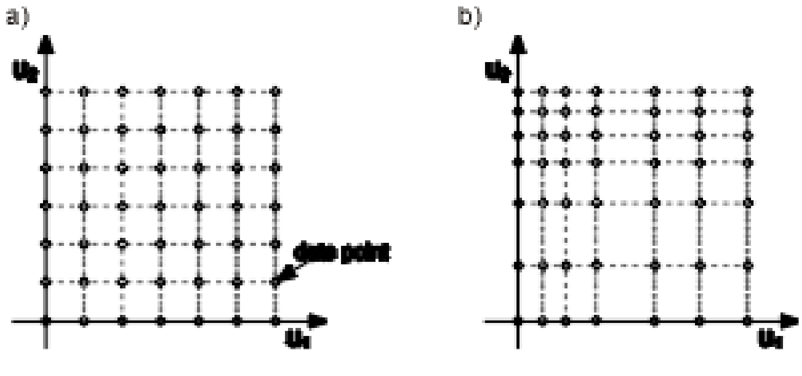

Figure 1: Lattice structure with data points for a) an equidistant grid and b) a non-equidistant grid

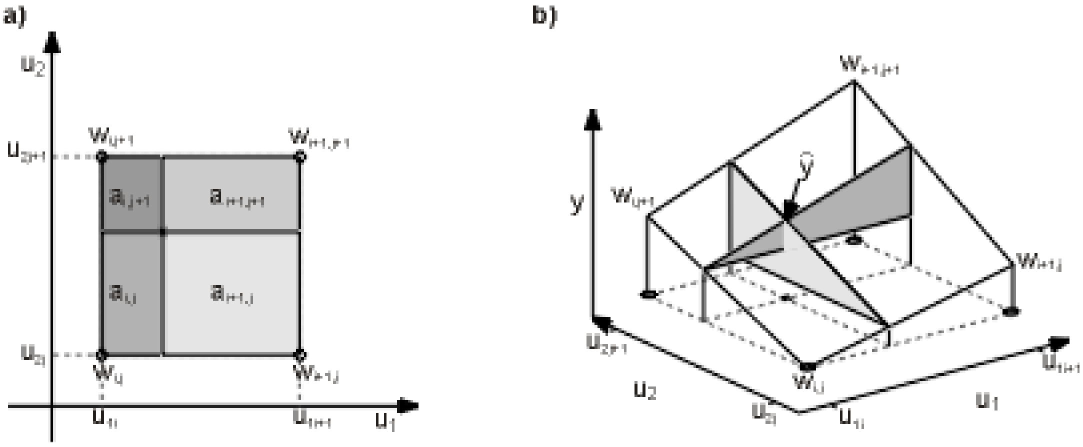

Figure 2: Bilinear interpolation as a) area interpolation rule and as b) recursive linear interpolation

While optimising look-up tables, usually all data point positions are fixed a-priori. Either an equidistant grid is chosen or due to physical insights any grid structure can be assumed. There exist two ways determining data point weights. The most widely applied possibility is to position measurement input data directly on the grid. Then the weights are given by averaged output measurements and no optimisation algorithm is necessary. The second way allows arbitrary measurement data distribution. Then the data point weights have to be optimised due to a linear optimisation method. If the lattice structure itself is subject to optimisation, nonlinear techniques have to be applied. Because of the constraints regarding the lattice structure, this optimisation is very difficult and seldom used in practice.
Due to the regular topology and the resulting simple address calculation, grid-based look-up tables are very well suited for the modelling of one- and two-dimensional nonlinear mappings. Indeed, extensions to more input variables lead to an exponentially growth of data point numbers. There total number is given by the equation:

$$N = \prod_{i=1}^{n} N_i \tag{2.2}$$

where N_i is the number of nodes in input direction i. Equation (2.2) illustrates the curse of dimensionality problem and consequently the given grid structure is not suited to deal with higher-dimensional input spaces. Furthermore, the uniform distribution of data points in the input space does not allow an arrangement of the nodes with respect to the different complexities of nonlinearity in different regions of the input space. In the next Section, a new look-up table model is introduced which overcomes this restrictions.

3 HIERARCHICAL LOOK-UP TABLES

Most nonlinear black-box models for approximation of static nonlinearities can be described as linear combinations of nonlinear basis functions

$$\hat{y}(\boldsymbol{u}) = \sum_{i=1}^{M} w_i \Phi_i(\boldsymbol{u}, \boldsymbol{v}_i) \tag{3.1}$$

where $\hat{y}$ is the scalar model output of the n-dimensional input $\boldsymbol{u}$. Different approaches differ in the chosen basis function Φ_i. These basis functions are referred to as tuneable basis functions if they can be placed to the data by varying $\boldsymbol{v}_i$. Contrary to this, conventional approaches use basis functions which are fixed a-priori. Such models include polynomial expansions, splines and Fourier expansions. Look-up tables can also be expressed in form of equation (3.1). Then each parameter w_i in (3.1) is equivalent to a data point weight and the basis functions Φ_i correspond to the area weighting due to (2.1). Consequently, the parameter vector $\boldsymbol{v}_i$ characterises a data point position. Usually for grid-based look-up tables a-priori given basis functions are utilised or at least all $\boldsymbol{v}_i$ are constraint to lie on a grid.
Hierarchical look-up tables permit the adjustment of their basis functions to fit the measurement data. That means, the data points can be placed with respect to the local complexity of the modelled function. Therefore, this models share the learning capability of ANN. The data points of this approach have not to lie on a grid. Nevertheless, no arbitrary positions are allowed. The hierarchical construction algorithm performs axis-orthogonal cuts and divides the input space into hypercubes of any size. Equivalent to the grid-based models the vertices of these hypercubes serve as data point positions.

3.1 CONSTRUCTION ALGORITHM

The optimisation of data point positions is a nonlinear structure identification problem. Hierarchical approaches represent an excellent way to deal with such complex problems. Here a hierarchical construction algorithm is proposed building up a binary tree-structured look-up table. The algorithm extends incrementally the look-up table by adding new data points. In each iteration step the hypercube with the worst approximation quality is axis-orthogonally partitioned into two smaller hypercubes. Their vertices define the new nodes of the look-up table. Simultaneously, the input space is recursively partitioned. Figure 3a shows possible resulting data point locations. The heuristic algorithm comprises the iterative execution of the following steps:

1. *Initialisation of the look-up table*:
   The 2^n data point positions of the initial model are defined by the 2^n vertices of the smallest hypercube surrounding all measurement data. Their corresponding data point weights are estimated with a least-squares algorithm (LS).
2. *Test of all decompositions with their new data point positions*:
   There exist n distinct axis-orthogonal directions for the decomposition of the actual hypercube in the n-dimensional input space, see Figure 3b. The following decomposition rules can be used:
   (a) The hypercubes are partitioned into two halves.
   (b) The hypercubes can be partitioned in any ratio and the best one has to be found by nonlinear optimisation.
   For each of the n possible splits, a maximum on 2^{n-1} new nodes results. All data point heights have to be reestimated with LS-algorithm and the loss function has to be calculated for all n decompositions.
3. *Selection of the best decomposition*:
   The decomposition from step 2, mostly decreasing the loss function value, is chosen for improvement of the look-up table model.
4. *Selection of the hypercube with the worst approximation quality*:
   A local error measure for each hypercube itself has to be calculated. The hypercube with the worst approximation is chosen for further refinement.
5. *Test of the termination criteria*:
   If the termination criteria is fulfilled, tree growing stops else go to step 2.

The hierarchical approach and the limitation to axis-orthogonal cuts allow the separation of the complex and nonlinear structure identification problem into easy-to-handle smaller pieces. In each iteration step the decomposition problem is solved in a combinatorial way by choosing the best splitting from a finite number of possible ones. Note that, for the applicability of the interpolation rules (2.1), the restriction to axis-orthogonal cuts is necessary.

Due to the proposed partitioning rules, data point distribution depends on the local complexity of the modelled function. That means, more nodes appear in regions of stronger nonlinearities than in regions where a nearly linear behaviour is given. While searching optimal data points, this algorithm performs an input space partitioning which is equivalent to the local linear model tree (LOLIMOT) algorithm (Nelles, 1999). This LOLIMOT algorithm constructs very efficiently local linear neuro-fuzzy models. Similar hierarchical input space decompositions are proposed in Murray-Smith and Johansen (1997). Other heuristic strategies for node positioning of look-up table models can be found in Ullrich and Tolle (1997) and Schmitt et. al. (1994) where Delaunay triangulation and scattered data interpolation are proposed.

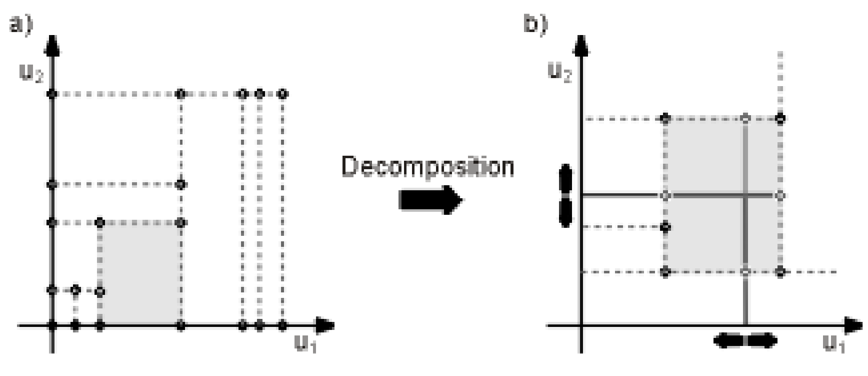

Figure 3: a) Hierarchical decomposition of the input space with corresponding data point positions and b) search for the best splitting of a hypercube

3.2 ACCESS TO HIERARCHICAL LOOK-UP TABLES

Look-up tables constructed by the hierarchical algorithm above are binary tree-structured. Accordingly, access to these models has to be performed by dropping down the tree. The terminal nodes of such a tree correspond to the distinct hypercubes of the input space. All branches describe the hierarchical decomposition of the input space where each non-terminal node characterises the splitting of the successor hypercube by it's direction and position. If a full tree structure is assumed, searching a hypercube is of computational complexity $O(\log_2(N))$. Similar to grid-based look-up tables, the hierarchical ones process inputs in a two-stage algorithm:

1. Selection of the enclosing nodes by tree search
2. Output interpolation

The first step comprises the selection of the 2^n data points surrounding the input vector. These nodes can be found very fast by searching the tree from top to bottom which is pictured in Figure 4. At each non-terminal node the respective variable of the input vector is compared with the split position. Consequently, the terminal nodes contain the address of the memory locations for the data point weights of the corresponding active data points. Only these data points contribute to the sum in (3.1) with non-zero basis functions. All other basis functions are zero. Once a hypercube with it's corresponding nodes is selected, the same interpolation rules (2.1) as for grid-based look-up tables are applied. Contrary to ANN where all neurons contribute with their nonlinear behaviour to the overall neural network output, only local information concerning the enclosing hypercube is required. This fact makes output calculation of look-up tables very fast, but the price to be paid is the non-differentiable transition between different interpolation areas.

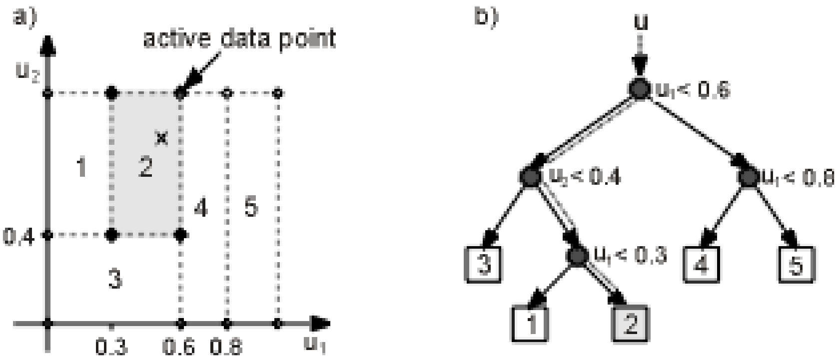

Figure 4: Selection of the enclosing nodes: a) input space with the active data points and b) searching the tree for the hypercube which encloses the input data

3.3 APPLICATION TO AN AUTOMOTIVE MODELLING PROBLEM

To illustrate the approach, the identification of a steady-state engine torque characteristic is investigated. Here a process with a two-dimensional input space is chosen to make visualisation of the partitioning and parsimonious location of data points in 3D maps possible. The engine torque is a nonlinear function of the throttle angle and the engine speed. A set of 433 measurement data shown in Figure 5a was generated on a combustion engine test stand. Simulation results for a hierarchical look-up table are illustrated in Figures 5b-d. The hierarchical algorithm was stopped after the creation of 12 hypercubes with 25 data points. Figure 5b shows the decomposition of the two-dimensional input space with the related positions of data points where decomposition of hypercubes was always performed into two halves. In the operating range characterised by high throttle angle values, only few measurement data are available. As can be seen from Figure 5b, only two interpolation rules were established by the algorithm in this area. Because of the lack of more detailed information in this part of the operating range, further refinement makes no sense and increases the risk of overfitting. Note that, least-squares optimisation of grid-based look-up tables without incorporation of this a-priori knowledge would fail, since several data point weights could not be determined.
The normalised root mean square error (NRMSE) for the training data is 0.084. The model quality is equivalent to the results achieved with a Multilayer Perceptron neural network with five neurons (total number of 21 parameters) in the hidden layer. Hence, hierarchical look-up tables are parsimonious models which can yield high generalisation quality.

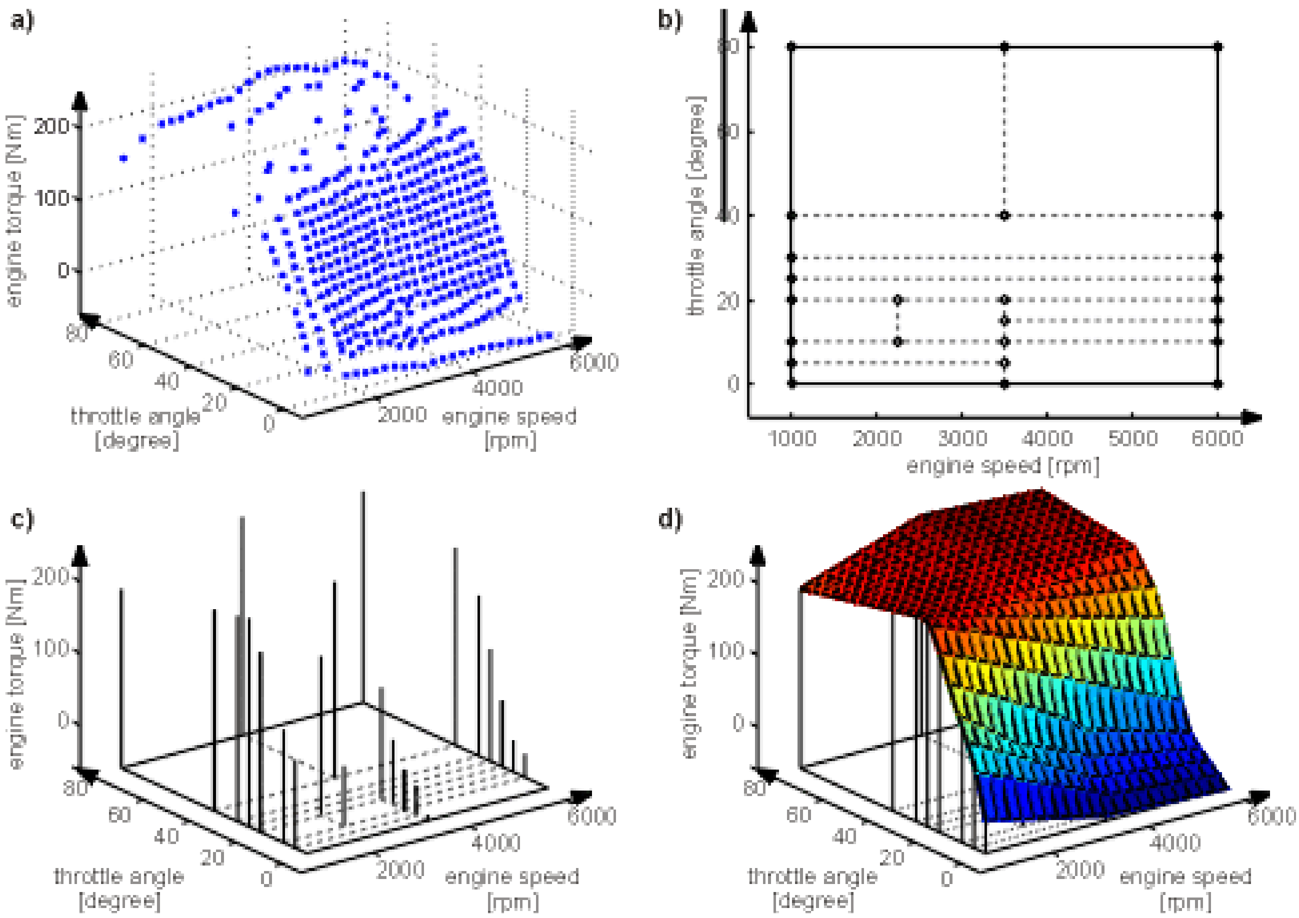

Figure 5: a) Measurement data and b) the decomposition of the input space with corresponding data point positions, c) data point weights and d) interpolation behaviour of the overall engine torque model

4 CONCLUSIONS

The concept of hierarchical look-up tables was pointed out. These models turn away from the rigid lattice structure to a more flexible data point distribution. The hierarchical construction algorithm of this approach inserts data points by a heuristic strategy with respect to the properties of the modelled function. Topological relations of the resulting input space decomposition are represented by a tree structure which allows fast access to the enclosing nodes of new input data. For given active data points the same interpolation rules as for lattice-like look-up tables are employed. The depth of look-up table trees is related to the refinement of the input space and does not directly depend from the input space dimensionality. Hence, the hierarchical look-up tables do not suffer from the curse of dimensionality problem and can be applied in systems with limited computational power.

REFERENCES

Breiman L.; Friedman J.; Ohlsen R.; Stone C.J., 1984, “Classification and Regression Trees”, Wadsworth Belmot, CA.

Murray-Smith R.; Johansen T. A., 1997, “Multiple Model Approaches to Modelling and Control”, Taylor & Francis

Nelles O., 1999, ”Nonlinear System Identification with Local Linear Neuro-Fuzzy Models”, PhD thesis, Technical University of Darmstadt, Shaker Verlag

Schmitt M.; Ullrich T.; Tolle H., 1994, “Associative Datafields in Automotive Control”, *3rd IEEE International Conference on Control Applications*, Glasgow (UK), pp. 1239-1244.

Schmitt M., 1995, “Untersuchungen zur Realisierung mehrdimensionaler lernender Kennfelder in Großserien-Steuergeräten”. PhD thesis, Technical University of Darmstadt, VDI-Verlag, Fortschrittsberichte Reihe 12 Nr. 246

Ullrich T.; Tolle H., 1997,”Modelling Nonlinear Characteristics using Hierarchical Delaunay-Networks”, *2-nd IMACS Conference on Mathematical Modelling*, Vienna (Austria), pp.769-774.