

# INCREMENTAL IDENTIFICATION OF NARX MODELS BY SPARSE GRID APPROXIMATION

Olaf Kahrs, Marc Brendel, Wolfgang Marquardt<sup>1</sup>

*Lehrstuhl für Prozesstechnik, RWTH Aachen University,  
D-52064 Aachen, Germany*

Abstract: Nonlinear empirical models are used in various applications. During model-building, five major steps usually have to be carried out: model structure selection, determination of input variables, complexity adjustment of the model, parameter estimation and model validation. These steps have to be repeated until a satisfactory model is found, which can be very time consuming and may require user interaction. This paper proposes an algorithm based on sparse grid function approximation to incrementally build a nonlinear empirical model. The algorithm exhibits good performance in terms of manual effort and computation time. The method is illustrated by a case study on the identification of a NARX model.  
*Copyright ©2005 IFAC*

Keywords: Nonlinear models, sparse grids, input selection, NARX

## 1. INTRODUCTION

Many important industrial processes exhibit nonlinear behavior. For model-based control and optimization a process model is required. Physically-based process models are often not available due to development and maintenance cost. Therefore an empirical process model may be identified from experimental or plant data as described by e.g. Henson and Seborg (1997).

As a typical example, the identification of discrete-time, nonlinear, auto-regressive models with exogenous inputs (NARX models) is considered in this paper, although the methodology presented can easily be applied to other model structures as well. The general formulation of a NARX model for a process with a single input  $u \in \mathbb{R}$  and a single output  $y \in \mathbb{R}$  is  $y_k = f(\mathbf{x}_k)$ , where  $f$  is a nonlinear function. The input variables of the NARX model

$$\mathbf{x}_k = [u_{k-1}, \dots, u_{k-p}, y_{k-1}, \dots, y_{k-q}] \quad (1)$$

are delayed samples of the process input and output, using the notation  $y_k = y(t = t_k)$ . The model order parameters  $p$  and  $q$  determine how many delayed samples are included in  $\mathbf{x}_k$ .

Given a measurement data set, the NARX model is usually built in an iterative process that consists of five major steps:

(1) *Select a model structure:* A suitable model structure for the function  $f$  has to be selected from the many alternatives that are proposed in literature, such as artificial neural networks or polynomial models (Henson and Seborg, 1997). The selection is usually strongly influenced by personal preferences and the available software but should be based on the intended application. For instance, Pearson (2003) states four important measures for model utility in process control: approximation accuracy, physical interpretability, ease of controller design and ease of model development.

---

<sup>1</sup> Corresponding author. E-mail: marquardt@ipt.rwth-aachen.de

(2) *Determine input variables:* The input variables of the NARX model, i.e. the model order, are often assumed to be known. Alternatively, they are determined by building multiple model candidates with different sets of input variables and selecting the best one. This time-consuming search strategy can be avoided by algorithms that suggest a model order prior to the choice of a model structure as was presented by e.g. Feil *et al.* (2004).

(3) *Adjust the model complexity:* The concepts of stepwise model refinement, model pruning, i.e. deleting dispensable parameters, and regularization are frequently employed when model complexity is adjusted, see e.g. Bishop (1995) for artificial neural networks. Model complexity is often expressed by the number of model parameters and is determined by the grade of nonlinearity in the data set as well as the desired model accuracy.

(4) *Estimate the model parameters:* A parameter estimation problem, which is sometimes also combined with step (3), is set up and solved by an algorithm that is usually tailored to the model structure for numerical efficiency.

(5) *Validate the identified model:* After a solution has been obtained, the model should be validated, in order to decide if some of the preceding steps have to be repeated to improve the model.

The model-building process can be very time-consuming if many model candidates with different sets of input variables and model complexities have to be considered. The goal of this paper is to reduce the repetitions of the model-building steps by a systematic model-building strategy that combines all steps into one algorithm. Brendel and Marquardt (2003) noted that sparse grid approximation may find use for structure selection of nonlinear discrete-time models. This is demonstrated in this work.

Yserentant (1986) proposed the sparse grid approach for multilevel-splitting of finite-element spaces. It was mainly used for the solution of PDEs (Griebel *et al.*, 1992). Recently the sparse grid approach has been introduced to the field of function approximation by Garcke *et al.* (2001) and Brendel and Marquardt (2003). In this paper the stepwise model refinement algorithm of Brendel and Marquardt (2003) is modified to systematically select input variables and adjust model complexity. Furthermore, an alternative regularization term is proposed.

Sparse grid approximation is described briefly in Section 2, which is the basis of the systematic model-building strategy presented in Section 3. This strategy is exemplified in a case study on NARX model identification and the resulting NARX model orders are compared to those found

by Feil *et al.* (2004). Finally, some conclusions are given in Section 5.

## 2. SPARSE GRID FUNCTION APPROXIMATION

Two basic statistical assumptions are commonly made if an output error model  $\tilde{y}_k = f(\mathbf{x}_k) + \epsilon_k$  is used in function approximation: (i) the model structure  $f$  is reasonably correct and (ii) the measurements of the input variables  $\mathbf{x}_k$  in the  $k$ -th experiment contain negligible errors. All measurement errors and the model mismatch are described by the error term  $\epsilon_k$ , which is assumed to have a normal distribution with zero mean and variance  $\sigma^2$ .

The objective is to identify the function  $f: \mathbb{R}^d \rightarrow \mathbb{R}$  of some function space  $V$  from measurement data of the output variable  $\tilde{y}$  and the input variables  $\mathbf{x}$ . This inverse problem may be ill-posed in the sense of Hadamard and therefore requires regularization (Engl *et al.*, 1996).

The function  $f$  is the solution of the regularized least-squares estimation problem

$$\min_{f \in V} \frac{1}{M} \sum_{k=1}^M (f(\mathbf{x}_k) - \tilde{y}_k)^2 + \lambda \Phi(f). \quad (2)$$

The first term measures the average approximation error of the function  $f$  to the measured values  $\tilde{y}_k$  and the second term represents a regularization term to handle the ill-posedness.

The following subsections briefly describe how the function  $f$  is defined in the sparse grid approach. Since the sparse grid solution is composed of multiple so-called full grid solutions, the function approximation on a full grid is presented first.

### 2.1 Function approximation on a full grid

The function  $f$  is represented in the full grid approach by a truncated basis function expansion

$$f_{\mathbf{l}}(\mathbf{x}) := \sum_{\mathbf{j} \in \Lambda} \theta_{\mathbf{l}, \mathbf{j}} \phi_{\mathbf{l}, \mathbf{j}}(\mathbf{x}) \quad (3)$$

where  $\theta_{\mathbf{l}, \mathbf{j}}$  are parameters to be estimated and  $\phi_{\mathbf{l}, \mathbf{j}}(\mathbf{x})$  are basis functions that are local with finite support in  $d$ -dimensional space. The basis functions  $\phi_{\mathbf{l}, \mathbf{j}}(\mathbf{x})$  are parameterized by the level multi-index  $\mathbf{l}$  and the position multi-index  $\mathbf{j}$ , which have  $d$  entries each, where  $d$  is the number of variables in  $\mathbf{x}$ . The level multi-index  $\mathbf{l}$  defines the mesh size  $h_{\mathbf{l}} = (h_{l_1}, \dots, h_{l_d}) = (2^{-l_1}, \dots, 2^{-l_d})$  of the full grid in each dimension. The position multi-index  $\mathbf{j}$  translates the basis function in each dimension to the coordinates  $\mathbf{x}_{\mathbf{l}, \mathbf{j}} = (j_1 2^{-l_1}, \dots, j_d 2^{-l_d})$  with  $\mathbf{j} \in \Lambda := \{\mathbf{j} \mid j_i \in \{0, \dots, 2^{l_i}\}, i = 1 \dots d\}$ . For

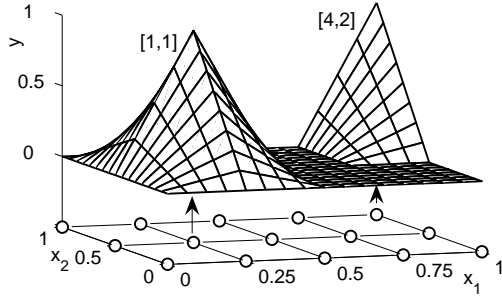


Fig. 1. Two basis functions for level  $\mathbf{l} = (2, 1)$  on the positions  $\mathbf{j} = (1, 1)$  and  $\mathbf{j} = (4, 2)$ .

convenience the input variables  $\mathbf{x}$  are transformed first to the interval  $[0, 1]^d$ .

The basis functions are created from the piecewise-linear mother function  $\phi(x)$  by dyadic dilation and translation:

$$\phi_{\mathbf{l}, \mathbf{j}}(\mathbf{x}) := \prod_{i=1}^d \phi_{l_i, j_i}(x_i), \quad (4)$$

$$\phi_{l_i, j_i}(x_i) := \phi\left(\frac{x_i - j_i h_{l_i}}{h_{l_i}}\right), \quad (5)$$

$$\phi(x) := \begin{cases} 1 - |x| & \text{if } x \in ]-1, 1[ \\ 0 & \text{otherwise} \end{cases}. \quad (6)$$

The basis functions are centered on the grid points of a grid with a uniform mesh size in each dimension. For instance, two basis functions are shown in Figure 1 for a two-dimensional full grid.

The sparse grid approach can be generalized to other basis functions including higher-order polynomials, interpolets, prewavelets and wavelets (Bungartz and Griebel, 2004), e.g. if  $f$  has to be continuously differentiable.

The mesh size determines how accurate the full grid can approximate a given data set. However, the finer the mesh the more parameters have to be estimated. The number of parameters of a full grid with level  $\mathbf{l}$  is  $N_{\mathbf{l}} = \prod_{i=1}^d (2^{l_i} + 1)$  and thus it increases exponentially with both the number of model inputs  $d$  and the level  $\mathbf{l}$ .

## 2.2 Replacement of the full grid by a sparse grid

To cope with the high number of parameters in the full grid approach, Griebel *et al.* (1992) proposed a combination technique for the solution of sparse grid problems. The solution on a large full grid is approximated by combining solutions on a set of smaller full grids, so-called subgrids. More precisely, the sparse grid solution  $f_{\mathbf{l}}^{(c)}$  on level  $\mathbf{l}$  is the weighted sum of all subgrid solutions on levels  $\mathbf{s}$  that are elements of an index set  $\mathcal{I}_{\mathbf{l}}$ :

$$f_{\mathbf{l}}^{(c)}(\mathbf{x}) := \sum_{\mathbf{s} \in \mathcal{I}_{\mathbf{l}}} w_{\mathbf{s}} f_{\mathbf{s}}(\mathbf{x}). \quad (7)$$

The superscript  $(c)$  denotes the combination of subgrid solutions. Each subgrid  $f_{\mathbf{s}}$  is a full grid on level  $\mathbf{s}$ , defined by equation (3) with the level index  $\mathbf{l}$  being replaced by  $\mathbf{s}$ .

Different subgrid combination techniques were presented by Garcke and Griebel (2002) and Brendel and Marquardt (2003). Brendel and Marquardt (2004) compare those techniques and state that the anisotropic grid combination technique from Garcke and Griebel (2002) seems to be favorable, therefore it is used in this work.

The index set for the anisotropic grid combination technique is defined by

$$\mathcal{I}_{\mathbf{l}} := \left\{ \mathbf{s} \mid \sum_{i=1}^d \frac{s_i}{l_i} \leq 1, s_i \geq 0 \right\}. \quad (8)$$

With its characteristic function

$$\chi^{\mathcal{I}_{\mathbf{l}}}(\mathbf{s}) := \begin{cases} 1 & \text{if } \mathbf{s} \in \mathcal{I}_{\mathbf{l}}, \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

the weights in equation (7) result in

$$w_{\mathbf{s}} := \sum_{\mathbf{z}=(0, \dots, 0)}^{(1, \dots, 1)} (-1)^{\|\mathbf{z}\|_1} \chi^{\mathcal{I}_{\mathbf{l}}}(\mathbf{s} + \mathbf{z}). \quad (10)$$

These weights take integer values. The associated subgrid solution is dispensable if a weight is zero. Table 1 gives an example for the subgrid decomposition of a sparse grid with level  $(4, 2, 2, 1)$  along with the number of parameters  $N_{\mathbf{s}}$  of each subgrid. A full grid on level  $(4, 2, 2, 1)$  would contain 1275 parameters, whereas the sparse grid is composed of 10 subgrids with 484 parameters in total. All subgrid solutions are calculated separately, thus the largest parameter estimation problem involves only 136 parameters.

Table 1. Subgrids on level  $\mathbf{l}=(4,2,2,1)$ .

$\mathbf{s}$	$w_{\mathbf{s}}$	$N_{\mathbf{s}}$	$\mathbf{s}$	$w_{\mathbf{s}}$	$N_{\mathbf{s}}$
(0, 0, 0, 1)	1	24	(0, 0, 1, 0)	-2	24
(0, 0, 2, 0)	1	40	(0, 1, 0, 0)	-2	24
(0, 1, 1, 0)	1	36	(0, 2, 0, 0)	1	40
(2, 0, 0, 0)	-2	40	(2, 0, 1, 0)	1	60
(2, 1, 0, 0)	1	60	(4, 0, 0, 0)	1	136

## 2.3 Estimation of sparse grid parameters

The sparse grid parameter estimation problem is easily solved, when it is expressed as a linear set of equations. Therefore, all parameters  $\theta_{\mathbf{s}, \mathbf{j}}$  and basis functions  $\phi_{\mathbf{s}, \mathbf{j}}(\mathbf{x})$  of a subgrid with level  $\mathbf{s}$  are arranged to column vectors  $\theta_{\mathbf{s}}$  and  $\phi_{\mathbf{s}}(\mathbf{x})$  using a bijective mapping function  $M : \mathbb{R}^d \rightarrow \mathbb{R}$  that maps the multi-index  $\mathbf{j}$  to an element of the column vector. Thus, equation (3) can be rewritten for a subgrid  $\mathbf{s}$  as  $f_{\mathbf{s}} = \phi_{\mathbf{s}}^T(\mathbf{x}_k) \theta_{\mathbf{s}}$  for a single record of the data set and as  $\mathbf{f}_{\mathbf{s}} = \mathbf{J}_{\mathbf{s}}^T \theta_{\mathbf{s}}$  for

the whole data set. The matrix  $\mathbf{J}_s$  is the Jacobian of  $\mathbf{f}_s$  w.r.t. to  $\theta_s$  with column  $k$  being  $\phi_s(\mathbf{x}_k)$ .

Garcke and Griebel (2002) investigated sparse grids in data mining applications, where the output variable  $\tilde{y}$  is a discrete valued class label. For regularization they choose a Tikhonov regularization term defined by  $\Phi(f_s) := \theta_s^T \mathbf{C}_s \theta_s$  with the regularization matrix  $\mathbf{C}_s$  defined elementwise by

$$(\mathbf{C}_s)_{v,w} := \int_{[0,1]^d} \nabla \phi_{s,v}^T(\mathbf{x}) \nabla \phi_{s,w}(\mathbf{x}) d\mathbf{x}. \quad (11)$$

For regression with  $\tilde{y} \in \mathbb{R}$  an alternative regularization term is proposed that enforces smoothness of the solution. Since the basis functions are d-linear, the smoothness is evaluated by summing up squared 2nd-order finite differences along all dimensions. The finite differences are evaluated at the basis function centers where the function values are equal to the corresponding parameter values. The regularization term is defined by

$$\Phi(f_s) := \sum_{i=1}^d \sum_{\mathbf{j} \in \Lambda_i} \left( \frac{\theta_{s,\mathbf{j}-\mathbf{e}_i} - 2\theta_{s,\mathbf{j}} + \theta_{s,\mathbf{j}+\mathbf{e}_i}}{2^{-2s_i}} \right)^2, \quad (12)$$

where the lower and upper border of the grid are taken out from finite difference calculation by defining  $\Lambda_i := \Lambda \setminus \{\mathbf{j} \mid j_i \in \{0, 2^{s_i}\}\}$ .  $\mathbf{e}_i$  denotes the  $i$ -th unit vector. Equation (12) can be rewritten in the form  $\Phi(f_s) := \theta_s^T \mathbf{D}_s \theta_s$  with a regularization matrix  $\mathbf{D}_s$ .

The sparse grid parameters are estimated by solving the linear problems

$$(\mathbf{J}_s \mathbf{J}_s^T + \lambda M \mathbf{D}_s) \theta_s = \mathbf{J}_s \tilde{\mathbf{y}} \quad \forall s, \quad (13)$$

which can be done in parallel.  $M$  is the number of data points in the estimation data set.

The regularization parameter  $\lambda$  can be chosen in different ways. Brendel and Marquardt (2003) apply cross-validation and the L-curve criterion from Hansen (1998) and state that cross-validation renders more reliable estimates of  $\lambda$ . Therefore, the regularization parameter is estimated by 10-fold cross-validation throughout this work. It should be noted here that the sparse grid solution is regularized in two ways. First, the chosen discretization level is an inherent regularization (Kirsch, 1996). Additionally, the Tikhonov term regularizes the solution, which becomes significant with increasing discretization level.

### 3. GRID REFINEMENT ALGORITHM

The preceding section presented the identification of a sparse grid for a given set of input variables  $\mathbf{x}$  and a specified discretization level  $\mathbf{l}$ . However, a suitable level is usually not known a-priori.

Brendel and Marquardt (2003) proposed a grid refinement algorithm to avoid an exhaustive search through all possibilities. This algorithm is extended for the purpose of input variable selection in the present work. Furthermore, the criterion for selecting the refinement step is modified.

For automatic input variable selection a discretization level of -1 is formally introduced. All input variables that are discretized on level -1 are simply ignored when calculating the sparse grid solution. For example  $f_{(4,-1,0)}^{(c)}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) = f_{(4,0)}^{(c)}(\mathbf{x}_1, \mathbf{x}_3)$ . If the level is -1 in all dimensions, all input variables are ignored. In this case, the function  $f$  is defined as the mean value of  $\tilde{\mathbf{y}}$ .

The optimal regularization parameter is estimated by cross-validation. The whole data set is split up in  $N_{cv}$  parts.  $N_{cv} - 1$  parts are used for estimating the parameters and the remaining part for validation of the solution. The superscript  $r$  denotes the  $r$ -th part, while the superscript  $[r]$  denotes all data sets except the  $r$ -th part. Accordingly, the parameter vector  $\theta_s^{[r]}$  is the solution of equation (13) on data set  $(\mathbf{x}^{[r]}, \tilde{\mathbf{y}}^{[r]})$ .

The cross-validation error

$$e_{cv}(\mathbf{l}, \lambda) := \frac{1}{M} \sum_{r=1}^{N_{cv}} \left\| \tilde{\mathbf{y}}^r - \sum_{s \in \mathcal{I}_1} w_s f_s(\mathbf{x}^r, \theta_s^{[r]}) \right\|_2^2 \quad (14)$$

is a function of the regularization parameter and its minimum yields the optimal regularization parameter

$$\hat{\lambda}(\mathbf{l}) := \arg \min_{\lambda} e_{cv}(\mathbf{l}, \lambda). \quad (15)$$

The grid refinement algorithm starts with an initial grid of level  $\mathbf{l} = \{-1, \dots, -1\}$ . Thereafter the grid is successively refined by the algorithm:

- (1) Calculate the cross-validation error  $e_{cv}(\mathbf{l}, \hat{\lambda}(\mathbf{l}))$  for the current grid  $\mathbf{l}$ .
- (2) Create a set of extended grids of levels  $\mathbf{l}_i = \mathbf{l} + \mathbf{e}_i$  with  $i = 1 \dots d$  by sequentially bisecting the current grid in one dimension.  $\mathbf{e}_i$  is the  $i$ -th unit vector.
- (3) For each extended grid calculate  $e_{cv}(\mathbf{l}_i, \hat{\lambda}(\mathbf{l}_i))$ .
- (4) Find the dimension  $\hat{i}$  that leads to the lowest cross-validation error.
- (5) If  $e_{cv}(\mathbf{l}_{\hat{i}}, \hat{\lambda}(\mathbf{l}_{\hat{i}})) > e_{cv}(\mathbf{l}, \hat{\lambda}(\mathbf{l}))$  or some other termination criterion is met, then the grid refinement is finished. Otherwise select the extended grid corresponding to  $\hat{i}$  as current grid ( $\mathbf{l}_{\hat{i}} \rightarrow \mathbf{l}$ ) and proceed with step (2).

During the refinement process equation (13) has to be solved several times for different values of  $\lambda$ . For numerical efficiency the matrices  $\mathbf{D}_s$  and  $\mathbf{J}_s$  are stored and reused.

## 4. EXAMPLE

The sparse grid refinement algorithm is exemplified in a case study on the identification of a NARX model. For this model class the set of input variables is usually unknown a-priori. Feil *et al.* (2004) demonstrated a clustering-based model order selection approach with data sets generated by a simulation model of a continuous polymerization reactor, which is also used for comparison in this work.

### 4.1 First-principles reactor model

The model describes the free-radical polymerization of methyl methacrylate with azobisisobutyronitrile as an initiator and toluene as a solvent. The reaction takes place in a jacketed continuous stirred tank reactor. Under some simplifying assumptions, the first-principle model is given by:

$$\begin{aligned}\dot{x}_1 &= 10(6 - x_1) - 2.4568x_1\sqrt{x_2} \\ \dot{x}_2 &= 80u - 10.1022x_2 \\ \dot{x}_3 &= 0.024121x_1\sqrt{x_2} + 0.112191x_2 - 10x_3 \\ \dot{x}_4 &= 245.978x_1\sqrt{x_2} - 10x_4 \\ y &= \frac{x_4}{x_3}.\end{aligned}$$

The model input  $u$  is the dimensionless volumetric flow rate of the initiator and the model output  $y$  is the number-average molecular weight. For further information on this model see (Doyle *et al.*, 1995).

A uniformly distributed random input  $u$  over the range 0.007 – 0.0015 with a sampling time of 0.2 s is applied to the model. The output  $y$  is scaled to zero mean and unit variance. White noise with a variance of 0.01 is added to  $y$  in order to make the subsequent NARX identification task more realistic. Delayed samples of  $u$  and  $y$  are assembled to two data sets: one for parameter estimation and one for validation, including 2000 and 1000 data points respectively.

### 4.2 Sparse grid NARX reactor models

A NARX one-step-ahead prediction model is identified from the simulated measurement data of the first-principles reactor model. The *incremental* model building approach is compared to a *classical* approach in terms of the mean square one-step-ahead prediction error  $e$  on the validation data set, the computation time  $T$  and the number of model parameters  $N^{(c)}$ . It is emphasized that the prediction error  $e$  is not the same as the cross-validation error  $e_{cv}$  that is used to estimate the optimal regularization parameter. The computation time  $T$  is measured in seconds

for the current implementation of the algorithm in Matlab (The MathWorks, Inc., 2004), which is executed on a personal computer (CPU: AMD Athlon XP 2600+, RAM: 1.5 GB). The reported levels correspond to the NARX input variables with the ordering defined in equation (1). For example, the level (2, 1, 0) with model orders  $p = 2$  and  $q = 1$  encode the NARX model  $y_k = f(u_{k-1}, u_{k-2}, y_{k-1})$ , where the sparse grid was discretized on level 2 for  $u_{k-1}$ , on level 1 for  $u_{k-2}$  and on level 0 for  $y_{k-1}$ .

In the *classical* model-building approach a set of NARX model candidates is identified with different model orders and a fixed discretization level. The model with the lowest prediction error on the validation data set is chosen as final model. The input variables to the NARX model have to be defined a-priori, thus all possibilities for selecting delayed process inputs  $u_{k-p}$  and delayed process outputs  $y_{k-q}$  in equation (1) are taken into account. Here  $p$  and  $q$  are limited to the interval [0, 3]. In general, different discretization levels can be used for each input variable of the NARX model. In this approach the level is fixed to 2, in order to narrow the set of possible model candidates. It is assumed that this level will ensure a sufficient flexibility of the sparse grid to approximate the data set.

The identification results were obtained after 71.9 s of CPU time and are presented in Table 2. The rigid discretization results in a high number of parameters. However, even for high  $N^{(c)}$  the prediction error is close to the variance of 0.01 of the noise that was added to  $y_k$ . This indicates that the Tikhonov regularization term efficiently prevents the models from overfitting.

The model with  $p = 2$  and  $q = 1$  and the level (2, 2, 2) is chosen as final model, because it has a low prediction error  $e$  as well as a low number of parameters  $N^{(c)}$ .

Table 2. Classical approach.

Input delays $p$		Output delays $q$			
		0	1	2	3
0	$e$	1.0044	0.7701	0.7560	0.7567
	$T$ [s]	0.1	0.4	0.4	0.9
	$N^{(c)}$	1	5	41	158
1	$e$	0.2576	0.0284	0.0117	0.0107
	$T$ [s]	0.1	0.4	1.1	2.5
	$N^{(c)}$	5	41	158	488
2	$e$	0.0249	0.0107	0.0107	0.0108
	$T$ [s]	0.4	1.1	2.4	8.0
	$N^{(c)}$	41	158	488	1392
3	$e$	0.0112	0.0108	0.0109	0.0110
	$T$ [s]	1.0	2.5	8.4	42.3
	$N^{(c)}$	158	488	1392	3760

The *incremental* model-building approach uses the grid refinement algorithm presented in Section 3. The initial level is set to -1 for automatic selection of suitable model inputs. Model complexity

is adjusted by the grid refinement algorithm to reflect the characteristics of the given data set. The maximal model order is limited to  $p = q = 3$  to conveniently present the results.

Table 3 shows the intermediate best grid refinement levels. The solution with level (2, 1, -1, 0, 0, -1) is obtained after only 10.4 s of CPU time (corresponding to  $p = q = 2$ ) in refinement step 8 with a prediction error of 0.0107. The refinement level (1, 0, -1, 0, -1, -1) in refinement step 5, which corresponds to  $p = 2$  and  $q = 1$ , is almost optimal in terms of the cross-validation error  $e_{cv}$ , while only requiring 12 parameters. The user may choose this model if simulation time is essential in the envisioned application.

Table 3. Incremental approach.

Step	Level l	$e_{cv}$	$N^{(c)}$
1	(-1, -1, -1, -1, -1, -1)	44.962	1
2	(0, -1, -1, -1, -1, -1)	0.2700	2
3	(0, 0, -1, -1, -1, -1)	0.0313	4
4	(0, 0, -1, 0, -1, -1)	0.0131	8
5	(1, 0, -1, 0, -1, -1)	0.0102	12
6	(2, 0, -1, 0, -1, -1)	0.0101	20
7	(2, 0, -1, 0, 0, -1)	0.0099	40
8	(2, 1, -1, 0, 0, -1)	0.0099	80

The approach of Feil *et al.* (2004) indicates, prior to selecting any model structure, a model order of  $p = 2$  and  $q = 1$ . However, this model order has still to be validated by identifying a limited number of model candidates. The results of the classical and the incremental approach show that the indicated model order is indeed suitable for this case study.

## 5. CONCLUSIONS

Stepwise refinement of a sparse grid was used to automatically select model input variables and to adjust model complexity in an efficient way without requiring any user interaction. The algorithm was exemplified in a case study on NARX model identification. The determined model order is validated by an exhaustive search and compares well to the model order reported by Feil *et al.* (2004).

Future research will focus on:

- higher-order basis functions to obtain continuously differentiable functions,
- locally adaptive sparse grids to further reduce the number of required parameters,
- an alternative refinement criterion that also takes the number of parameters into account,
- an input design strategy that is integrated in the stepwise model refinement procedure to optimize experimental data acquisition,
- more complex case studies on multi-input multi-output (MIMO) processes.

## ACKNOWLEDGEMENTS

This work was partially supported by the German Federal Ministry for Science and Education (BMBF) under grant 03C0315A.

## REFERENCES

- Bishop, C.M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press. New York.
- Brendel, M. and W. Marquardt (2003). Stepwise refinement of sparse grids in data mining applications. In: *Proc. of the 13th IFAC Symposium on System Identification, SYSID 2003*. Rotterdam, The Netherlands. pp. 1667–1672.
- Brendel, M. and W. Marquardt (2004). A function approximation algorithm based on hierarchically refined sparse grids. *In preparation*.
- Bungartz, H.-J. and M. Griebel (2004). Sparse grids. *Acta Numerica* **13**, 1–123.
- Doyle, F.J., B.A. Ogunaike and R.K. Pearson (1995). Nonlinear model-based control using second-order volterra models. *Automatica* **31**(5), 697–714.
- Engl, H. W., M. Hanke and A. Neubauer (1996). *Regularization of Inverse Problems*. Kluwer Academic Publishers.
- Feil, B., J. Abonyi and F. Szeifert (2004). Model order selection of nonlinear input-output models – a clustering based approach. *J. Proc. Cont.* **14**(6), 593–602.
- Garcke, J. and M. Griebel (2002). Classification with sparse grids using simplicial basis functions. *Intelligent Data Analysis* **6**(6), 483–502.
- Garcke, J., M. Griebel and M. Thess (2001). Data mining with sparse grids. *Computing* **67**, 225–253.
- Griebel, M., M. Schneider and C. Zenger (1992). A combination technique for the solution of sparse grid problems. In: *Iterative Methods in Linear Algebra* (P. de Groen and R. Beauwens, Eds.). Elsevier. Amsterdam. pp. 263–281.
- Hansen, P. C. (1998). *Rank-deficient and Discrete Ill-posed Problems*. SIAM. Philadelphia.
- Henson, M. A. and D. E. Seborg (1997). *Nonlinear Process Control*. Prentice Hall. New Jersey.
- Kirsch, A. (1996). *An Introduction to the Mathematical Theory of Inverse Problems*. Springer. New York.
- Pearson, R. K. (2003). Selecting nonlinear model structures for computer control. *J. Proc. Cont.* **13**(1), 1–26.
- The MathWorks, Inc. (2004). *MATLAB, Release 14*.
- Yserentant, H. (1986). On the multi-level splitting of finite-element spaces. *Numerische Mathematik* **49**, 379–412.