15th Triennial World Congress, Barcelona, Spain

# DESIGNING STATE-SPACE MODELS WITH NEURAL NETWORKS

**Józef Korbicz** * **Marcin Mrugalski** * **Thomas Parisini** **

* *Institute of Control and Computation Engineering*
*University of Zielona Góra*
*ul. Podgórna 50, 65–246 Zielona Góra, Poland*
*fax: +48 68 3254615,*
*e-mail: {J.Korbicz,M.Mrugalski}@issi.uz.zgora.pl*
** *Departament of Electrical, Electronic and Computer Engineering*
*DEEI-University of Trieste*
*Via Valerio 10 34127 Trieste, 20133 Milano, Italy*
*Italy Fax: +39 040 6763460,*
*e-mail: parisini@univ.trieste.it*

Abstract: This paper presents a new state-space identification framework for non-linear systems. In particular, a state-space model structure is designed with the Group Method of Data Handling type neural network. It is assumed that the neurons of the network have tangensoidal activation functions. For such a network type, a new approach based on a bounded-error set estimation technique is employed to estimate the parameters of the network. The final part of this work contains an illustrative example regarding the application of the proposed approach in the fault detection system.

Keywords: State-space models, neural networks, non-linear system identification, fault diagnosis

## 1. INTRODUCTION

The complexity and reliability demands of contemporary industrial systems and technological processes requires the development of new fault diagnosis approaches. The early detection of faults may help to avoid a system breakdown and material damages. During the last few decades many investigations have been made using analytical approaches, based on mathematical models. One of the most known structures of the model-based fault diagnosis system is based on the residual generation. An application of models leads directly to the problem of system identification. One way out of this problem is application of Artificial Neural Networks (ANNs) (Duch *et al.*, 2000). The attractiveness of ANNs follows from the fact that they are useful when there are no phenomenological models available, i.e. the models which are built with the physical consideration underlying the system of interest. Such a situation causes that behavioural models, i.e. the models which merely approximate the observed behaviour, should be employed. In this case, the model structure does not claim to correspond in any more way to that of the system and the parameters of the model have no physical meaning. Moreover, if it is impossible to approximate the system behaviour accurately using linear models, then the ANN seems to be an appropriate tool. The direct identification of state-space models has recently attracted research attention (Van Overschee and De Moor, 1994). In general the model designing can be viewed as the problem of finding a mapping between the available input-output data sequences and unknown parameters in a user defined class of state-space models. Recent surge of interest follows from some unique advantages of state-space identification over the input-output based approaches. A state-space description can be numeri-

cally better conditioned then an input-output description. Moreover, state-space model construction can be readily extended to multi-input, multi-output system. Thus, it seems especially attractive to develop a system identification framework for non-linear system with the application of ANNs.

Unfortunately, in spite of the considerable usefulness of ANNs, there are no efficient algorithms for selecting structures of the ANNs and hence many experiments should be carried out to obtain an appropriate configuration. In addition to that, the parameter (weights) estimation problem is formulated usually as a non-linear optimization one. Among the existing algorithms, a few groups can be distinguished: gradient-based algorithms (Hertz, Krogh and Palmer, 1991), evolutionary algorithms (Angeline, 1995; Koza, 1992; Michalewicz, 1996), stochastic algorithms (Walter and Pronzato, 1997). Unfortunately, the training of the ANN is usually an optimization problem of a multimodal cost function. This means, that the gradient-based algorithms usually find one of the unsatisfactory local minima. To overcome this problem, it seems desirable to use either stochastic or evolutionary algorithms, possess global convergence properties. Unfortunately, the number of parameters of an ANN is rather large, this leads to an increase in the computational burden, which makes the parameter estimation process extremely time consuming. On the other hand, there are many works which confirm the applicability of the GMDH *(Group Method of Data Handling)* approach (Farlow, 1984; Ivakhnenko and Muller, 1995). In this case, the structure of the network is successively increased by adding new neurons, according to the prespecified criterion. The main advantage of this approach is that the problem boils down to identifying the parameters of a single neuron only (problem decomposition). This means that, if the activation function is invertible then the problem reduces to linear parameter estimation one. In this work, an ANN whose neurons have tangensoidal activation functions is considered. For such a network type, a state-space identification scheme is proposed on a new approach based on bounded-error set estimation (Fogel and Huang, 1982; Maksarow and Norton, 1996a; Maksarow and Norton, 1996b; Milanese *et al.*, 1996) is employed to the parameter estimation.

The paper is organized as follows. Section 2 presents some elementary information concerning fault diagnosis. In Section 3 some basic notation and the model representation used throughout the paper are described. Section 4 presents the structure of the GMDH type neural network. In Section 5 a concept of parameter estimation via bounded-error estimation set approach is described in detail. Moreover, it is shown how to modify the above algorithm in order to attain an automatic selection of the „appropriate" identification data set. The final part of this work contains an illustrative example, which confirms the effectiveness of the proposed approach.

## 2. MODEL-BASED FAULT DIAGNOSIS

One of the best-known structures of the fault diagnosis system is based on a model of a system being considered (Patton Frank and Clark, 2000) (Fig. 1). Model-based fault diagnosis can be defined as the detection, isolation and identification of faults in the system based on a comparison of the system's available measurements, with information represented by the system's mathematical model (Patton and Chen, 1999). Sometimes there are no mathematical description of the diagnosed system or the high complexity of real systems makes it impossible to perform it. In such situations, the behavioural models, which merely approximate the system input-output behaviour, have to be employed. One way out of this problem is the application of GMDH type neural networks (Duch *et al.*, 2000).

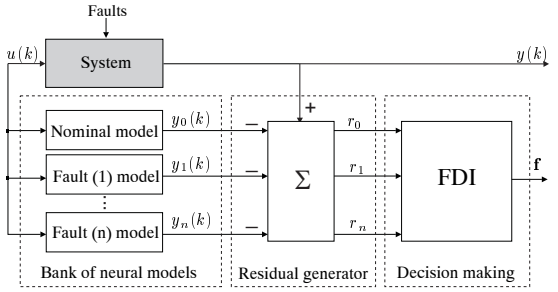


Fig. 1. The structure of a model-based FDI system

Using available input and output signals from the system, the objective is to generate a bank of models. The bank consist of GMDH neural models of the system under nominal conditions and some faulty states. Output signals estimated by the bank of models output are used to generate the difference between the model and system outputs. Such a difference signal is usually called a residual and can be employed as a source of information for *Fault Detection and Isolation* (FDI). The residual signal should be normally close to zero in the fault free mode, otherwise it should be distinguishably different from zero when a fault occurs. The algorithm used to generate residuals is called a residual generator. The residual should ideally carry only fault information. Faults are detected by setting a fixed or variable threshold on a residual signal. A number of residuals can be designed, each a heaving special sensitivity to individual faults occurring in different locations in the system. The decision making part can be realized based on artificial intelligence approach such as neural networks or fuzzy-logic.

## 3. MODEL DESCRIPTION

Let us consider the following class of non-linear discrete-time systems

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{a}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{w}_k, \\ \boldsymbol{y}_{k+1} &= \boldsymbol{c}\boldsymbol{x}_{k+1} + \boldsymbol{v}_k. \end{aligned} \tag{1}$$

Thus, the state-space model of the system (1) can be expressed as

$$\begin{aligned} \hat{\boldsymbol{x}}_{k+1} &= \boldsymbol{a}(\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k), \\ \hat{\boldsymbol{y}}_{k+1} &= \boldsymbol{c}\hat{\boldsymbol{x}}_{k+1}. \end{aligned} \tag{2}$$

where $\boldsymbol{x}_k, \hat{\boldsymbol{x}}_k \in \mathbb{R}^n$ are the state vector and its estimate, $\boldsymbol{u}_k \in \mathbb{R}^m$ is the input vector, $\boldsymbol{y}_k, \hat{\boldsymbol{y}}_k \in \mathbb{R}$ are the observed output vector and its estimate, $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are the process and measurement noises. The objective is to obtain the vectors $\boldsymbol{a}(\cdot)$ and $\boldsymbol{c}$, given a set of input-output measurements. Moreover, it is assumed the true state vector $\boldsymbol{x}_k$ is, in particular, unknown. Without loss of generality, it is possible to assume that

$$\begin{aligned} \boldsymbol{a}(\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k) = &[a^1(\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k), \\ &a^i(\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k, a^{i-1}(\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k)), \\ &\dots, \\ &a^n(\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k, a^{n-1}(\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k))]. \end{aligned} \tag{3}$$

Thus, the problem reduces to identifying non-linear functions $a^i(\cdot), i = 1, \dots, n$, and the vector $\boldsymbol{c}$. The identification of non-linear functions $a^i(\cdot)$ can easily be achieved by the following neuron structure

$$a^i(\cdot) = \xi({\mathbf{r}_k^i}^T \mathbf{p}), \quad i = 1, \dots, n, \tag{4}$$

where $\mathbf{r}_k^i = (\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k)$ for $i = 1$, and $\mathbf{r}_k^i = (\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k, a^{i-1}(\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k))$ for $i = 2, \dots, n$ is the $k$–th input vector, $\mathbf{p} \in \mathbb{R}^{n_p}$ is the parameter vector and $\xi(\cdot)$ is a hyperbolic tangent function. It should be also pointed out that the order $n$ of the model is in general unknown and hence should be determined throughout experiments. Moreover, each entry of the state vector, i.e. $\hat{x}_k$ represents, in one way or another, the output of the system.

$$x_k^i = y_k \quad \text{for} \quad i = n, \tag{5}$$

$$x_k^i = xk^i + \delta_k^i, \quad \text{for} \quad i = 1, \dots, n-1, \tag{6}$$

where

$$\delta_k^i = \xi'({\mathbf{r}_k^i}^T \mathbf{p}) \sum_{j=1}^{n_j} p_{k,j}^{i+1} \delta_{k,j}^{i+1}, \tag{7}$$

This means that each entry of the vector $\boldsymbol{a}(\cdot)$ should be obtained in such a way that each of the states should be well-suited to the system output. After the network designing procedure, knowing the state estimate and using the least-square method it is possible to obtain the vector $\boldsymbol{c}$ by solving the following equation

$$\boldsymbol{c} = \sum_{k=1}^{n_t} \boldsymbol{y}_k \hat{\boldsymbol{x}}_k^T \left[ \sum_{k=1}^{n_t} \hat{\boldsymbol{x}}_k \hat{\boldsymbol{x}}_k^T \right]^{-1}. \tag{8}$$

## 4. SYSTEM IDENTIFICATION USING GMDH NETWORKS

Successful identification depends on a proper selection of the model structure. In the case of ANNs, the problem reduces to the selection of the number of layers and the number of neurons in a particular layer. If the network obtained does not satisfy prespecified requirements, then a new network structure is selected and the parameter estimation problem is repeated once again. Thus, it seems desirable to have a tool, which can be employed to the automatic selection of the ANN structure, based only on the measured data. One way to solve this problem is to use the so called GMDH type neural networks (Fig. 2). Based on the

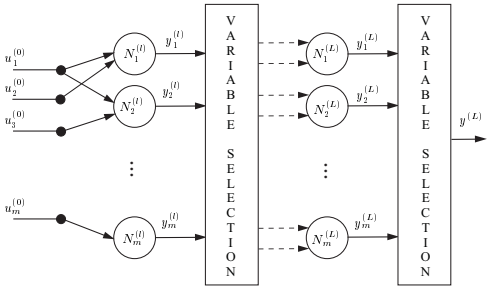


Fig. 2. Synthesis of the GMDH type neural network

number of available external inputs $(u_1, ..., u_m)$ , the network grows its first layer of the neurons. The output of each neuron is usually a combination of its inputs. The activation function of the neuron is assumed to be a tangensoidal function. The output of the neuron may become the input to other neurons in the next layer. During the training the GMDH network the number of layers increases. Each time when a new layer is added new neurons are introduced. The selection of best performing neurons for their processing accuracy is realized before the formed layer is added to the network. The remaining neurons are removed. There exist a few methods of performing the selection procedure:

- *Constant population method* is based on the selection of $g$ neurons, for which $E(y_m^{(l)})$ reaches the least values.
- *Optimal population method* is based on the rejection the neurons for which the defined quality index is larger than an arbitrarily determined threshold $e_h$ (Fig. 3). Usually $e_h$ is determined separately for each layer.

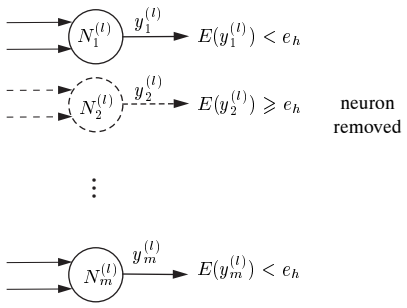


Fig. 3. The neuron selection in the layer

- *Decreasing population method* defines the maximum number of elements in a layer. The number of the neurons in each layer decreases along with the growth of the network.

In order to perform the model construction procedure it is necessary to define the quality index. In this work, this quality index has the following structure

$$E(y_m^{(l)}) = \frac{\sum_{k=1}^{n_t} (y_{m,k}^{(l)} - y_k)^2}{\sum_{k=1}^{n_t} y_k^2}, \tag{9}$$

where $n_t$ is the number of validation data, $y_k$ is the $k$-th scalar measurement of the system output, however $y_{m,k}^{(l)}$ is the neuron output. Only well performing neurons are preserved to build a new layer. Finally, when additional layers do not improve the performance of the network, the synthesis process is stopped. To avoid the problem of overfitting the identification data, the experimental input-output data is divided into two sets, namely the identification and validation sets. This means that the first set is used to estimate the parameters of the neurons, and the second one is used to select best performing neurons. To obtain the final structure of the network, all unnecessary neurons are removed, leaving only those which are relevant to the computation of the model output.

## 5. PARAMETER ESTIMATION VIA BOUNDED-ERROR SET ESTIMATION APPROACH

Let us reconsider the neuron structure

$$y_{m,k}(\mathbf{p}) = \mathbf{r}_k^T \mathbf{p} \tag{10}$$

$$y_{n,k}(\mathbf{p}) = \xi(y_{m,k}), \tag{11}$$

Using the fact that $\xi(\cdot)$ is an invertible activation function, the parameter estimation problem can be solved by means of the well developed linear parameter estimation approaches (Walter and Pronzato, 1997). Indeed, the output error can be defined as

$$e_{y,k}(\mathbf{p}) = y_{z,k} - y_{m,k}(\mathbf{p}), \tag{12}$$

where

$$y_{z,k} = \xi^{-1}(y_k), \tag{13}$$

$y_k$ is the $k$–th scalar measurement of the system output, and $y_{m,k}(\mathbf{p})$ is the corresponding neuron output. The usual statistical parameter estimation framework assumes that the data are corrupted by the errors which can be modeled as realizations of independent random variables, with a known or parameterized distribution. The estimator is usually designed based on the well-known least-squares criterion which results in the least-squares, or recursive least-squares, algorithm. A more realistic approach is to assume that the errors lie between given prior bounds, this is the case, for example, for the data collected with an analogue-to-digital converter or for measurements performed with a sensor of a given type. In this case, the output error is assumed to be bounded as follows

$$e_{y,k}^m \le e_{y,k}(\mathbf{p}) \le e_{y,k}^M, \tag{14}$$

where the bounds $e_{y,k}^m$ and $e_{y,k}^M$ are known a priori and $e_{y,k}^m \neq e_{y,k}^M$.These bounds may come from empirical knowledge, or simply intuition. The inequalities (14) associated with k-th measurement can be put into the standard form

$$-1 \le \bar{y}_{z,k} - \bar{y}_{m,k}(\mathbf{p}) \le 1, \tag{15}$$

with

$$\bar{y}_{z,k} = \frac{2y_{z,k} - e_{y,k}^M - e_{y,k}^m}{e_{y,k}^M - e_{y,k}^m}, \tag{16}$$

$$\bar{y}_{m,k}(\mathbf{p}) = \frac{2}{e_{y,k}^M - e_{y,k}^m} y_{m,k}(\mathbf{p}). \tag{17}$$

Let $\mathbb{O}_k$ be a strip in parameters space $\mathbb{P}$, bounded by two parallel hyperplanes. These hyperplanes are defined by

$$\mathbb{O}_k = \{\mathbf{p} \in \mathbb{R}^{n_p} : -1 \le \bar{y}_{z,k} - \bar{y}_{m,k} \le 1\} \tag{18}$$

In recursive outer-bounding ellipsoids algorithm (OBE) (Walter and Pronzato, 1997), the data are taken into account one after the other to construct a succession of ellipsoids containing all values of $\mathbf{p}$ consistent with all previous measurements. After the first $k$ observations, $\mathbf{p}$ that represents the set of feasible parameters is characterized by the ellipsoid

$$\begin{aligned}\mathbb{E}(\hat{\mathbf{p}}^k, \mathbf{M}_k) = &\Big\{\mathbf{p} \in \mathbb{R}^{n_p} : (\mathbf{p} - \hat{\mathbf{p}}^k)^T \\ &\mathbf{M}_k^{-1}(\mathbf{p} - \hat{\mathbf{p}}^k) \le 1\Big\},\end{aligned} \tag{19}$$

where $\hat{\mathbf{p}}^k$ is the center of the new ellipsoid consisting $k$-th parameter estimate, and $\mathbf{M}_k$ is a positive-definite matrix witch specifies its size and orientation. By means of intersection of the above strip and the ellipsoid, we get region of possible parameter estimates. This region is overbouned, by a new ellipsoid. The algorithm described below provides rules for computing $\mathbf{p}^k$ and $\mathbf{M}_k$ in such a way that

$$\mathbb{E}(\hat{\mathbf{p}}^{k+1}, \mathbf{M}_{k+1}) \supseteq \mathbb{E}(\hat{\mathbf{p}}^k, \mathbf{M}_k) \cap \mathbb{O}_k. \tag{20}$$

ensuring that the volume of $\mathbb{E}(\hat{\mathbf{p}}^{k+1}, \mathbf{M}_{k+1})$ is minimal. The center of the $n_t$-th, ellipsoid constitutes the resulting parameter estimate, where $n_t$ is the number of data points.

A detailed structure of this recursive algorithm is as follows:

(1) Select an initial estimate $\hat{\mathbf{p}}^0$, set $\mathbf{M}_0 = c\boldsymbol{I}$ ($c$ is a suefficiently large positive real number).

(2) While $\mathbf{r_k} = \mathbf{0}$, set

$$\mathbb{E}(\hat{\boldsymbol{p}}^k, \boldsymbol{M}_k) = \mathbb{E}(\hat{\boldsymbol{p}}^{k-1}, \boldsymbol{M}_{k-1}),$$
$$k \leftarrow k + 1.$$

(3) Calculate

$$y_{m,k} = \boldsymbol{r}_k^T \hat{\boldsymbol{p}}^k,$$

and normalize $y_{m,k}$ according to (16,17).

(4) Set

$$v = y_{z,k} - y_{m,k}, \quad g = \boldsymbol{r}_k^T \boldsymbol{M}_{k-1} \boldsymbol{r}_k.$$

(5) Calculate

$$a_+ = \max\left(\tfrac{v-1}{\sqrt{g}}, -1\right),$$

$$a_- = \max\left(-\frac{v+1}{\sqrt{g}}, -1\right).$$

(6) Update $v$ and $g$ to

$$v = \frac{a_- - a_+}{a_- + a_+}, \quad g = \left(\frac{2}{a_- + a_+}\right)^2.$$

(7) If $a_- > 1$ or $a_+ > 1$, then terminate the algorithm as conditions (14) are contradictory.

(8) If $a_+ a_- > 1/2$, then set $k \leftarrow k+1$, and go to step 3.

(9) Calculate

$$\alpha_1 = g^2,$$

$$\alpha_2 = g^2\left[\frac{1}{2}(a_+^2 + a_-^2) + 2a_- a_+ - 1 + \frac{2v^2}{g}\right],$$

$$\alpha_3 = g^2\left(\frac{a_- + a_+}{2}\right)(2a_+ a_- - 1),$$

$$\lambda = \frac{-\alpha_2 + \sqrt{\alpha_2^2 - 4\alpha_1\alpha_3}}{2\alpha_1},$$

$$c_1(\lambda) = 1 + \lambda - \frac{\lambda v^2}{1+\lambda g}.$$

(10) Set

$$\bar{\boldsymbol{M}}_k = \left[\boldsymbol{I} - \frac{\boldsymbol{M}_{k-1}\boldsymbol{r}_k\boldsymbol{r}_k^T}{\lambda^{-1} + \boldsymbol{r}_k^T\boldsymbol{M}_{k-1}\boldsymbol{r}_k}\right]\boldsymbol{M}_{k-1}.$$

(11) Update $\hat{\boldsymbol{p}}$ and $\boldsymbol{M}$:

$$\hat{\boldsymbol{p}}^k = \hat{\boldsymbol{p}}^{k-1} + \lambda v\bar{\boldsymbol{M}}_k\boldsymbol{r}_k,$$

$$\boldsymbol{M}_k = c_1(\lambda)\bar{\boldsymbol{M}}_k.$$

(12) If $k = n_t$, then STOP else $k \leftarrow k+1$, and go to step 2.

In spite of the considerable usefulness of the above algorithm. It is possible to increase its effectiveness even further. In particular, due to the fact that the matrix $\mathbf{M}^{-1}$ describes the parameters confidence region, it can be employed to the automatic selection of samples which provide a useful information in the sense of parameter identification. This means that it can be used to reject the samples which lead to an increase (in the sense of the criterion selected) in the size of the parameters confidence region. In order to perform such a modification, it is necessary the define an appropriate optimality criterion. In this work, it is assumed that the size of the ellipsoid (the parameters confidence region) is represented by the trace of the matrix $\mathbf{M}^{-1}$ (A-optimality criterion)

$$\Phi(M) = \min \text{trace}\left(\mathbf{M}^{-1}\right), \tag{21}$$

although it is possible to use other optimality criteria, e.g. D-optimality, E-optimality (Uciński, 1999). Thus, the main objective is to develop an approach which prevents trace $\left(\mathbf{M}^{-1}\right)$ increasing. It can be easily achieved by rejecting the samples which increase trace $\left(\mathbf{M}^{-1}\right)$.

## 6. SIMULATION EXAMPLE

The purpose of the present section is to show the application effectiveness of the proposed approach in the designing FDI system. In particular, the real data from an industrial plant were employed to identify the state-space model of chosen parts of the plant. The plant to be considered is the evaporation station at the Lublin Sugar Factor S. A. (Edelmayer, 2000). Fig. 4 shows the scheme of the plant with all the available process variables. Based on the observation of the

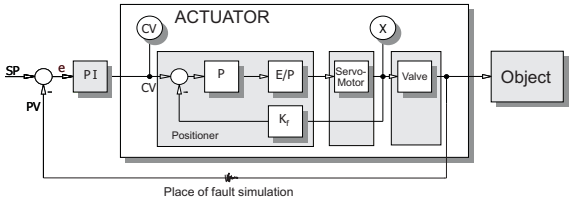


Fig. 4. A scheme of the system being considered

process variables and the knowledge about the process model can be expressed as:

$$P_v = f(C_v, X), \tag{22}$$

where $f(\cdot)$ denotes the modelled relationship, $P_v$ is the juice flow at the outlet of the evaporator, $C_v$ is the control value, $X$ represent servomotor rod displacement. The data used for the identification and validation sets were collected in November 2001. The data were, of course, appropriately filtered, moreover offset levels were removed with the use of the MATLAB identification toolbox. It should be also pointed out that these data sets were appropriately scaled for the purpose of neural networks designing. The output data signals should be transformed taking into consideration the response range of the output neurons. For the hyperbolic tangent neurons this range is $[-1, 1]$. To perform such kind a transformation, the linear scaling can be used. Moreover, to avoid saturation of the activation function, the raw output data was transformed to the interval $[-0.8, 0.8]$. The selection of best performing neurons for their processing accuracy is realized with application of the optimal population method approach. For the sake of comparison, the ARX model was obtained. In both the ARX and nonlinear GMDH state-space models cases the order of the model was tested between $n = 1, \ldots, 4$. Experimental results have shown that the best suited both models are of order $n = 4$. The quality indexes (9) for the ARX and GMDH state-space type model which is achieved with application of the modified OBE algorithm are given in Tab 1.

Table 1: Quality indexes for the identification and validation data sets

| | Identification | Validation |
|---|---|---|
| ARX | 1.169 | 1.464 |
| GMDH | 0.033 | 0.169 |

The comparative study performed shows that the GMDH model is superior to the ARX model. From this results it can be seen that the introduction of

the non-linear model has significantly improved modelling performance. The fault in the actuator was introduced in the control loop. The residual signal was be determined by a comparison of the measured values and model output. An occurrence of fault is signalled by a deviation of the residual value from zero. For this reason minimization of the identification error in the fault-free mode, as in the case of the proposed GMDH state-space model, is of a great importance. The residual signal for the GMDH state-space model is shown in Fig. 5. As can be seen the fault is very easy to detect., e.g. using a simple threshold technique.

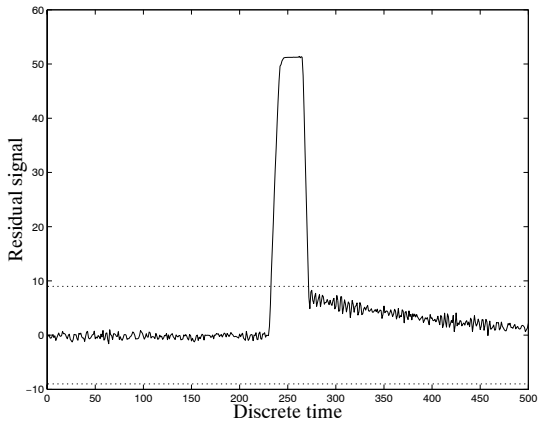


Fig. 5. Residual signal

## 7. CONCLUSIONS

The present work describes a new state-space model identification framework which is based on the GMDH neural networks. In particular, a comprehensive description of the network structure was presented and a suitable training algorithm was given. The main advantage to the proposed technique is that the parameter estimation problem can be formulated as a linear in parameter one. This makes this algorithm fast and effective. Another objective of this work was to develop a residual generation scheme based on the presented identification framework. The proposed identification technique was applied to the identification of a chosen part of an evaporation at the Lublin sugar factory. Finally, the resulting model was used to design a residual generation scheme. The reliability of the proposed technique was tested using artificially generated faults, i.e. the faults were introduced in the control loop. Experimental results confirm that the faults under consideration can be detested in a very straightforward way.

## ACKNOWLEDGEMENTS

This work was supported by the EU FP 5 Research Training Network project DAMADICS and in part by the State Committee for Scientific Research (KBN) in Poland.

## 8. REFERENCES

Angeline, P. (1995). Evolution Revolution: An introdution to the special track on genetic and evolutionary programing. *IEEE Expert*, **5**, pp. 6-10.

Duch, W., J. Korbicz, L. Rutkowski and R. Tadeusiewicz (2000). *Biocybernetyka i inynieria biomedyczna 2000. Sieci neuronowe*. Akademicka Oficyna Wydawnicza Exit, Warszawa. Tom 6. (in Polish).

Edelmayer, A. (2000). Research of Quantitative and Qualitative FDI Methods Based on Data From Lublin Sugar Factory. Spec. Session *IFAC Symp. SAFEPROCESS'2000*. Prepr. *IFAC Symp.: Fault Detection, Supervision and Safety of Technical Processes: SAFEPROCESS'2000*, June 14-16, Budapest, Hungary, pp. 331-358.

Farlow, S.J. (1984). *Self-Organizing Methods in Modeling - GMDH Type Algorithms*. Marcel Dekker, New York.

Fogel, E. and Y.F. Huang (1982). On the Value of Information in System Identyfication-Bounded Noise Case. *Automatica*, **18**(2), pp.229-238.

Hertz, J., R. Krogh, and G. Palmer (1991). *Introduction to the neural computation*. Addison-Wesley Publishing Company Inc.

Ivakhnenko, A.G. and J.A. Muller (1995). Self-Organizing of Nets of Active Neurons. *System Analysis Modelling Simulation*, **20**, pp. 93-106.

Koza, J.R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MA: The MIT Press, Cambridge.

Maksarow, D. and J. Norton (1996). Tuning noise bounds in state bounding. Proc. *CESA'96 IMACS Multiconference*, July 9-12, Lille, France, pp. 837-842.

Maksarow, D. and J. Norton (1996). State bounding with ellipsoidal set description of the uncertainty. *Int. Journal of Control*, **65**(5), pp. 847-866.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Berlin.

Milanese, M., J. Norton, H. Piet-Lahanier and E. Walter (Eds.) (1996). *Bounding Approaches to System Identification*. Plenum Press, New York.

Patton, R. J. and Chen J. (1999). *Robust Model-Based Fault Diagnosis For Dynamic Systems*. Kluwer Academic Publ, London.

Patton, R. J., Frank P. and Clark R.N. (Eds.) (2000). *Issues of Fault Diagnosis for Dynamic Systems*. Springer-Verlag, Berlin.

Uciński, D. (1997). *Measurement Optimization for Parameter Estimation in Distributed Systems*. Technical University Press, Zielona Góra.

Walter, E. and L. Pronzato (1997). *Identification of Parametric Models from Experimental Data*. Springer, London.

Van Overschee, P. and De Moor B. (1994). N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, **30**(1), pp. 75-93.