IFAC
Publications
www.elsevier.com/locate/ifac

# FAULT DETECTION WITH DYNAMIC GMDH NEURAL NETWORKS: APPLICATION TO THE DAMADICS BENCHMARK PROBLEM

**Marcin Mrugalski * Eugen Arinton * Józef Korbicz ***

* *Institute of Control and Computation Engineering,*
*University of Zielona Góra,*
*ul. Podgórna 50, 65–246 Zielona Góra, Poland,*
*e-mail: {M.Mrugalski,E.Arinton,J.Korbicz}@issi.uz.zgora.pl*

Abstract: This paper presents a relatively new identification method based on artificial neural networks, which can be used for both static and dynamic systems. In particular, a Group Method of Data Handling (GMDH) neural network with dynamic neurons is considered. The final part of the paper shows how to use the proposed approach to tackle fault detection of the DAMADICS benchmark. *Copyright © 2003 IFAC*

Keywords: fault diagnosis, non-linear systems, system identification, neural networks

## 1. INTRODUCTION

The complexity and reliability demands of contemporary industrial systems and technological processes require the development of new fault diagnosis approaches. The early detection of faults may help to avoid system breakdowns 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 a model-based fault diagnosis system is based on residual generation (Patton *et al.*, 2000). It is clear, that application of models leads directly to the problem of system identification. One way out of this problem is to apply Artificial Neural Networks (ANNs) (Nelles, 2001). The attractiveness of the 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 of the underlying 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. However, neural networks despite the small number of assumptions in comparison to statistical methods require a significant amount of *a priori* information about the model structure. Moreover, there are no efficient algorithms for selecting structures of classical ANNs and hence many experiments should be carried out to obtain an appropriate configuration. Experts should decide on the quality and quantity of input arguments, the number of layers and neurons as well as the form of their activation function. The heuristic approach that follows the determination of the questionable network architecture corresponds to a subjective choice of the final model, which in the majority of the cases will be a choice. To tackle this problem the GMDH approach can be employed (Ivakhnenko and Muller, 1995). The concept of this approach is based on the iterative processing of an operation defined sequence leading to the evolution of the resulting network structure. The GMDH network is a combination of the black box nonlinear system identification concept, and the inductive and probabilistic approaches. The assumptions of GMDH neural networks give a lot of freedom in defining the particular elements of the synthesis algorithm. The mentioned possibilities relate to, for example, the definition of the transition function, evaluation criteria of processing accuracy or selection methods. Another problem arises from the identification of dynamic systems. In the case of GMDH networks, it is impossible to apply well-known approaches for the classical neural network, which relies on the application of the static neural network with the global output feedback lines. It follows from the fact that the introduction of the global output feedback lines complicates the synthesis of the GMDH network. One way to avoid this problem is to use dynamic neurons (Korbicz *et al.*, 1999). The feed-forward structure of the network composed of the dynamic neurons seems to make the synthesis of the network process easier. In this work, an ANN whose dynamic neurons have hyperbolic tangent activation functions is considered. For such a network type, the classical least square method is employed to the parameter estimation. As an example, which confirms the effectiveness of the proposed approach in the system identification and fault detection tasks, the DAMADICSC benchmark problem is considered.

The paper is organized as follows. Section 2 presents the concept of the GMDH network synthesis, in particular the structure of the GMDH network is presented. In Section 3 the problem of modelling dynamics in GMDH networks is considered. In order to justify the reliability of the presented approaches, a comprehensive simulation study regarding the benchmark problem (DAMADICS, 2002) is performed in Section 4.

## 2. SYNTHESIS OF THE GMDH NEURAL NETWORK

The concept of the synthesis of the GMDH network is based on the iterative processing of a defined sequence of operations leading to the evolution of the resulting structure, which generates the output signal and is the best approximation of the real system output. The process is completed when the optimal degree of complexity is achieved.

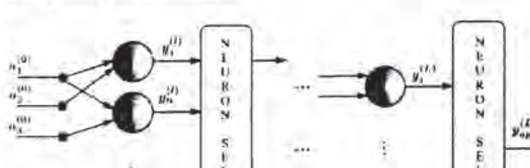


Fig. 1. Synthesis of the GMDH neural network

The GMDH neural network is constructed through the connection of a given number of elementary models (neurons) with the application of the appropriate selection methods (Fig. 1). The main characteristic of the GMDH method is the separate parameter estimation of each neuron. The parameters of the neurons are estimated in such a way that their output signals are the best approximation of the real system output. The training proceeds with introducing an input pattern to the input layer and adapting the weights of each neuron according to a suitable learning algorithm. It is assumed that at least two input signals $u_1^{(l)},\dots,u_m^{(l)}$ constitute the stimulation which results in the formation of the neuron output signal $y_n^{(l)}$:

$$y_n^{(l)} = f(\boldsymbol{u}) = f(u_1^{(l)},\dots,u_m^{(l)}), \qquad (1)$$

where $n = 1,\dots,N$, and $N$ is the number of neurons in the layer, $l = 1,\dots,L$, and $L$ denote the number of layers. The GMDH approach allows much freedom in defining of an elementary model transfer function $f$. The original GMDH algorithm developed by Ivakhnenko is based on the linear transfer functions or second-order polynomial functions (Ivakhnenko and Muller, 1995). From the practical point of view, (1) should be not too complex because it may complicate the learning process and extend the computation time. The number of neurons $N$ in the first layer of the network depends on the number of external inputs $m$. In a general case, a network of $m$ inputs is built from neurons that have $p$ inputs ($m > p$). In this case, $N$ new elements are formed in the following way:

$$N = \binom{m}{p} = \frac{m!}{p!(m-p)!}. \qquad (2)$$

The first layer of the GMDH network, for $p = 2$, has the structure shown in Fig. 2,

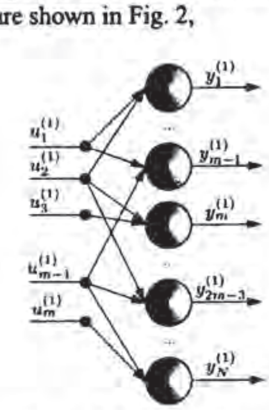

Fig. 2. First layer of the GMDH network

where

$$\begin{cases} y_1^{(1)} &= f(u_1^{(1)},u_2^{(1)}) \\ & \dots \\ y_{m-1}^{(1)} &= f(u_1^{(1)},u_m^{(1)}) \\ y_m^{(1)} &= f(u_2^{(1)},u_3^{(1)}) \\ & \dots \\ y_{2m-3}^{(1)} &= f(u_2^{(1)},u_m^{(1)}) \\ & \dots \\ y_N^{(1)} &= f(u_{m-1}^{(1)},u_m^{(1)}) \end{cases} \qquad (3)$$

The definition of the evaluation criterion $Q(y_n^{(l)})$ of the neurons is a preliminary task in designing a GMDH network (Mueller and Lemke, 2000). It allows any neuron to define the quantity of a processing error. Moreover, based on the defined evaluation criterion it is possible to make the selection of neurons in the layer. The selection of best performing neurons (Mrugalski *et al.*, 2002) in terms their processing accuracy is realized before the formed layer is added to the network. The parameters of the neurons in the newly created layer are "frozen" during the further network synthesis. The outputs of the selected neurons become the inputs to other neurons in the next layer:

$$\begin{cases} u_1^{(l+1)} = y_1^{(l)} \\ u_2^{(l+1)} = y_2^{(l)} \\ \dots \\ u_m^{(l+1)} = y_N^{(l)} \end{cases} \tag{4}$$

In analogous way, the new neurons in the next layers of the network are created. During the synthesis of the GMDH network, the number of layers suitably increases. Each time when a new layer is added, new neurons are introduced. The synthesis of the GMDH network is completed when the *optimum criterion* is obtained. The idea of this criterion is the determination of the quality index $Q(y_n^{(l)})$ for all $N$ neurons included in the $l$ layer. The $Q_{min}^{(l)}$ represents the processing error for the best neuron in this layer

$$Q_{min}^{(l)} = \min_{n=1,\dots,N} Q(y_n^{(l)}). \tag{5}$$

The values $Q(y_n^{(l)})$ can be determined with the application of the defined evaluation criterion used in the selection process. The values $Q_{min}^{(l)}$ are calculated for each layer in the network. The *optimum criterion* is obtained when the following condition occurs:

$$Q_{opt}^{(L)} = \min_{l=1,\dots,L} Q_{min}^{(l)}. \tag{6}$$

The $Q_{opt}^{(L)}$ represents the processing error for the best neuron in the network, which generate the model output signal. In other words, when additional layers do not improve the performance of the network, the synthesis process is stopped. To obtain the final structure of the network (Fig. 3), all unnecessary neurons are removed, leaving only those which are relevant to the computation of the model output. The procedure of

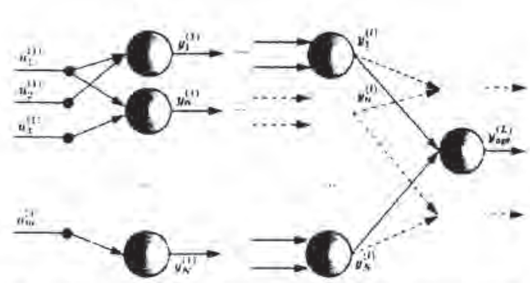

Fig. 3. Final structure of the GMDH neural network

removing unnecessary neurons is the last stage of the synthesis of the GMDH neural network.

## 3. DYNAMICS IN GMDH NEURAL NETWORKS

The most of the industrial systems are dynamic in its nature, and hence during identification it seems desirable to employ the models, which can represent the dynamic of the system. In the case of the classical neural network, like MultiLayer Perceptron (MLP), the problem of modelling of the dynamics is solved by introduction additional input signals. The input vector for this kind of networks consists of suitably delayed input and output signals. The input vector for this kind of networks consist of suitably delayed input and output signals

$$\begin{aligned} y(k) = & f(\boldsymbol{u}(k), \boldsymbol{u}(k-1), \dots, \boldsymbol{u}(k-n_u), \\ & y(k-1), \dots, y(k-n_y)), \end{aligned} \tag{7}$$

where $n_u$ and $n_y$ represents the number of delay in the input and output signals. According to the configuration of an identification system, it is possible to specify two identification structures, parallel and series-parallel (Fig. 4), in which the delayed output of the model $y_m$ or of the system $y$ is taken into account.

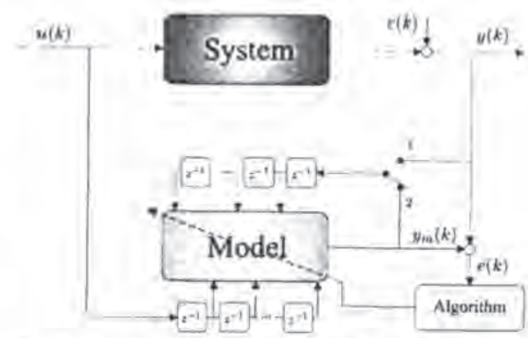


Fig. 4. Parallel and serial-parallel structures of the identification

In the case of the parallel identification structure, it is very difficult to assure the stability of the model obtained during identification. In other words, even though signals $u$ and $y$ are limited, the identification error $e$ must not be convergent on zero. From this reason, much more often applied is the series-parallel structure. Unfortunately, this kind of structure cannot be used in the fault detection task, where the model is used to generate the residual signal. If we assume that the identification error converge on zero, then the output signals of the model and the system can be described by the following relation

$$y = y_m. \tag{8}$$

From this reason, it is possible to use the parallel architecture during the final stage of the identification process, and an application in the fault diagnosis task. Unfortunately, the described approach cannot be applied into the GMDH neural network easily, because the GMDH network is constructed through connection of the elementary models. The introduction of global output feedback lines complicates the synthesis of the network. On the other hand, each neuron in the GMDH network constitutes an elementary model, which generates the output signal similar to the model output. In this situation, the elementary model should have an ability to represent the dynamics. One way out of this problem is to use dynamic neurons (Korbicz *et al.*, 1999). Due to introduction of different kinds of the local feedback to the classical neuron model, it is possible to achieve a few types of dynamic neurons (Korbicz *et al.*, 1999). The most known architectures are:

- Neurons with local activation feedback

$$y(k) = f\left(\sum_{i=1}^{p} w_i u_i(k) + \sum_{j=1}^{n_d} c_j \varphi(k-j)\right). \quad (9)$$

- Neurons with local synapse feedback

$$y(k) = f\left(\sum_{i=1}^{p} G_i(z^{-1}) u_i(k)\right), \quad (10)$$

where

$$G_i(z^{-1}) = \frac{\sum_{j=0}^{n_u} b_j z^{-1}}{\sum_{j=0}^{n_y} a_j z^{-1}}. \quad (11)$$

- Neurons with output feedback

$$y(k) = f\left(\sum_{i=1}^{p} w_i u_i(k) + \sum_{j=1}^{n_d} d_j y(k-j)\right). \quad (12)$$

The main advantage of networks constructed with application of the dynamic neurons is that their stability can be proven relatively easily. As a matter of fact, stability of the network depends only on stability of neurons. The feed-forward structure of such networks seems to make the training process easier. On the other hand, introduction of dynamic neurons increases the parameter space significantly. This drawback together with the non-linear and multi-modal properties of an identification index imply that parameter estimation becomes relatively complex.

On the other hand, there is another interesting way of introducing dynamic properties into the GMDH network with application of the other kind of a dynamic neuron model (Mrugalski *et al.*, 2002). Dynamics in this neuron is realized by introduction of a linear dynamic system - an Infinite Impulse Response (IIR) filter. In this way, each neuron in the network reproduces the output signal based on the past values of its inputs and outputs. Such a neuron model (Fig. 5) consists of two submodules: the filter module and the activation module. The behaviour of the filter module is described by the following equation:

$$\begin{aligned}\tilde{y}(k) = &- a_1\tilde{y}(k-1) - \ldots - a_{n_d}\tilde{y}(k-n_d) \\ &+ v_{1,0}u_1(k) + v_{2,0}u_2(k) +, \ldots, \\ &+ v_{m,0}u_m(k) \\ &+ v_{1,1}u_1(k-1) + v_{2,1}u_2(k-1) \\ &+, \ldots, + v_{m,1}u_m(k-1) \\ &\ldots \\ &+ v_{1,n_d}u_1(k-n_d) + v_{2,n_d}u_2(k-n_d) \\ &+, \ldots, + v_{m,n_d}u_m(k-n_d),\end{aligned} \quad (13)$$

or, equivalently,

$$\begin{aligned}\tilde{y}(k) = &- a_1\tilde{y}(k-1) - \ldots - a_{n_d}\tilde{y}(k-n_d) \\ &+ \boldsymbol{v}_0^T\boldsymbol{u}(k) + \boldsymbol{v}_1^T\boldsymbol{u}(k-1) +, \ldots, \\ &+ \boldsymbol{v}_{n_d}^T\boldsymbol{u}(k-n_d),\end{aligned} \quad (14)$$

where $a_1, a_2, \ldots, a_{n_d}$ denote the feedback filter parameters, $n_d$ represent the filter order, $\boldsymbol{v}_0^T, \boldsymbol{v}_1^T, \ldots, \boldsymbol{v}_{n_d}^T$ are the input vectors to the filter module, and $m$ is the number of inputs to the dynamic neuron. The filter output is used as the input for the next module, namely, the activation module. The activation module can be described by

$$y(k) = \xi(\tilde{y}(k)). \quad (15)$$

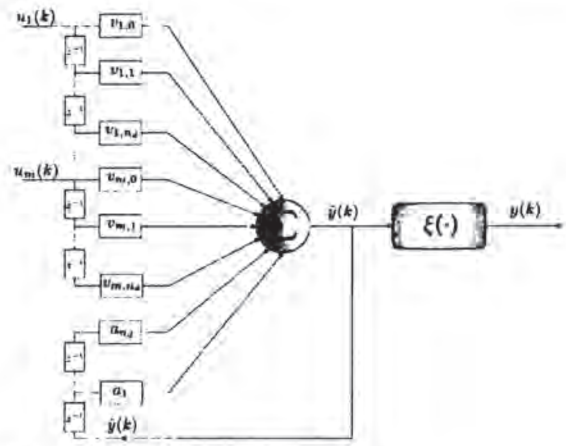


Fig. 5. A dynamic neuron model

The main objective is to estimate the parameters $a_1, a_2, \ldots, a_{n_d}$ and the parameter vectors $\boldsymbol{v}_i^T$, $i = 0, \ldots, n_d$. 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 (Korbicz *et al.*, 2002).

## 4. SIMULATION EXAMPLE

The purpose of the present section is to show the effectiveness of the proposed approach in the system identification and fault detection schemes. The Research Training Network on *Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems* (DAMADICS) is focused on the diagnosis of valve plant actuators and looks towards real implementation methods for new actuator systems, and in particular, on the diagnosis of a 5-stage evaporation and boiler station process of the Lublin Sugar Factory S.A. in Poland (DAMADICS, 2002). Three actuators have been chosen for research purposes. Two actuators connected with the evaporation station, the first one situated on the inflow of thin juice into the evaporation station, and the second one situated on the outlet of thick juice from the fifth section of the evaporation section. The third actuator connected with the fourth boiler house is situated on the water inflow into boiler drum. The element selected for modelling and fault detection, the third actuator, is a final control element, which interacts with the controlled process. The technological installation diagram with all available process variables is shown in Fig. 6.

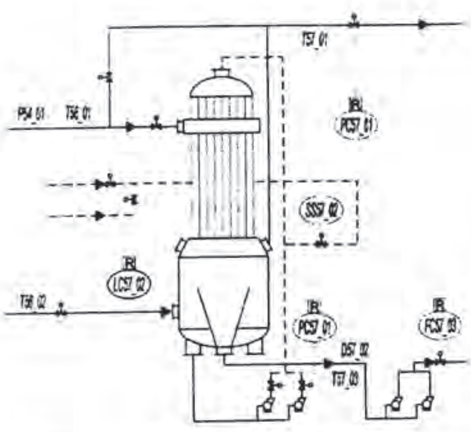

Fig. 6. Diagram of the diagnosed section of evaporation station

The input of the actuator is the output of the process controller (flow or level controller) and the actuator modifies the position of the valve allowing a direct effect on the primary variable in order to follow the flow or level set-point. Each actuator is equipped with similar measuring devices shown in Fig. 7, where $V_B$

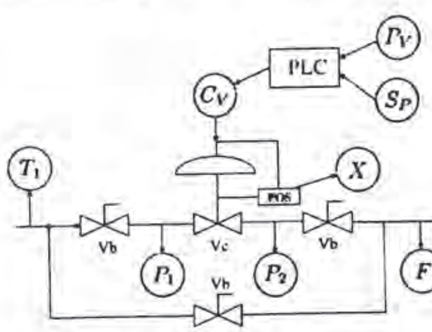


Fig. 7. Diagram of the actuator

and $V_C$ denote, the bypass and control valve, POS and PLC are the positioner and programmable logic controller. On the ground of the process analysis and taking into account the expert process knowledge, the following model of the juice flow at the outlet of the valve $F = r_F(X, P_1, P_2, T_1)$, and the servomotor rod displacement $X = r_X(C_V, P_1, P_2, T_1)$ were considered, where $r_F(\cdot)$ and $r_X(\cdot)$ denotes the modelled relationships, $C_v$ is the control valve, $P_1$ and $P_2$ are the pressures at the inlet and the outlet of the valve, respectively, $T_1$ represents the juice temperature at the inlet of the valve. The data used for the identification and validation sets were collected on 4th November 2001. It should be also pointed out that these data sets were 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 in terms of their processing accuracy was realized with the application of the optimal population method (Mrugalski *et al.*, 2002). The evaluation criterion $Q(y_n^{(l)})$ based on the regularity criterion (Mueller and Lemke, 2000) was defined where the identification data set was separated in two parts, training data set, and selection data set. The quality indexes for the training and validation sets, can be defined in the same following way $SSE_{t,v} = \sum_{k=1}^{n_{t,v}} (y(k) - y_{opt}^{(L)}(k))^2$. For the sake of comparison, the classical MLP, static and dynamic GMDH models were obtained. The dynamic GMDH network were realized with the application of dynamic neurons with the IIR filter. The order of the IIR filter was tested between $n_d = 1, \ldots, 5$. The experimental results showed that the best-suited models are of order $n_d = 2$. The quality indexes for models, obtained with the application of the least square method, are given in Tab 1.

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

| Neural network | $SSE_t$ | $SSE_v$ |
|---|---|---|
| MLP | 0.191 | 0.650 |
| Static GMDH | 0.231 | 0.259 |
| Dynamic GMDH | 0.163 | 0.166 |

The response of the system and the dynamic GMDH model obtained for the validation data set is given in Fig. 8. The comparative study performed shows that

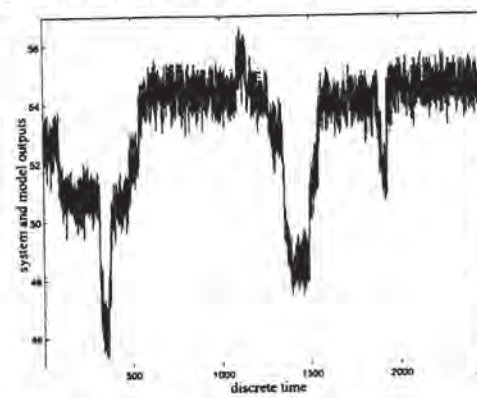


Fig. 8. System (—) and model (···) outputs

the dynamic GMDH model is superior to the MLP and static GMDH model. From the results presented in Tab 1. it can be seen that the introduction of the dynamic neurons has significantly improved modelling performance. Also for the fault detection real data from the sugar factory with faults introduced on 17th November 2001 was utilized. The residual signals for the fault $f_{17}$ – unexpected pressure drop across the valve, were determined by the comparison of the measured values and models outputs $X$ and $F$, respectively. The occurrence of a fault is signaled by the significant deviation of the residual from zero ($t = 50$–$85$). For this reason the minimization of the identification error in the fault-free mode, as in the case of the proposed dynamic GMDH model, is of a great importance. The residual signal for the dynamic GMDH model are shown in Fig. 9. As can be seen the fault is very easy to detect, e.g. using a simple threshold technique.

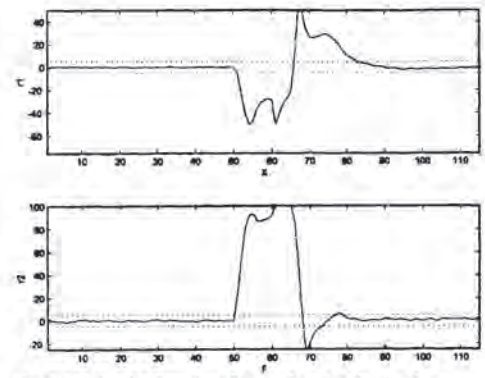

Fig. 9. Residuals for the fault $f_{17}$ (rod displacement – top, juice flow – botton)

Another objective was to provide a detailed description of an industrial application study of the proposed fault diagnosis scheme. For that purpose, based on a Simulink actuator model, 19 fault scenarios were simulated. To model suitable system outputs under normal condition the dynamic GMDH models were designed. In particular, model of the juice flow at the outlet of the valve $F = r_F(\cdot)$, and model of the servomotor rod displacement $X = r_X(\cdot)$ were constructed. Tab. 2 shows the results of fault detection. It should be pointed out that abrupt and incipient faults were considered. As can be seen, the abrupt faults presented in Tab. 2 can be considered as small, medium, and big according to the benchmark description (DAMADICS, 2002). The notation given in Tab. 2 can be explained as follows: $ND$ - means that it is impossible to detect a given fault, $DM_F$ and $DM_X$ - means that it is possible to detect a fault from the residual signal generated by model $r_F(\cdot)$ and $r_X(\cdot)$, respectively. From Tab. 2 it can be seen that it is impossible to detect faults $f_5, f_9, f_{12}$ and $f_{14}$. Indeed, the effect of this fault is exactly at the same level as the effect of noise. The residual is the same as that for fault-free case and hence it is impossible to detect this fault. There are also some problems with few small and medium abrupt faults, however, it should be pointed out that all faults (except for $f_{12}$ and $f_{14}$) which are considered as big can be detected.

Table 2: Results of fault detection

| Fault | Description | Abrupt | | | Incipient |
|---|---|---|---|---|---|
| | | Small | Med. | Big | |
| $f_1$ | Valve clogging | $DM_X$ | $DM_{F,X}$ | $DM_X$ | |
| $f_2$ | Valve plug or valve seat sedimentation | | | $DM_F$ | $DM_F$ |
| $f_3$ | Valve plug or valve seat erosion | | | | $DM_F$ |
| $f_4$ | Increased of valve or bushing friction | | | | $DM_X$ |
| $f_5$ | External leakage (leaky bushing, covers) | | | | $ND$ |
| $f_6$ | Internal leakage (valve tightness) | | | | $DM_F$ |
| $f_7$ | Medium evaporation or critical flow | $DM_{F,X}$ | $DM_X$ | $DM_X$ | |
| $f_8$ | Twisted servo-motor's piston rod | $ND$ | $ND$ | $DM_X$ | |
| $f_9$ | Servo-motor's housing or term. tightness | | | | $ND$ |
| $f_{10}$ | Servo-motor's diaphragm perforation | | $DM_X$ | $DM_X$ | |
| $f_{11}$ | Servo-motor's spring fault | | | $DM_X$ | $DM_X$ |
| $f_{12}$ | Electro-pneumatic transducer fault | $ND$ | $ND$ | $ND$ | |
| $f_{13}$ | Rod displacement sensor fault | $DM_F$ | $DM_F$ | $DM_F$ | $DM_F$ |
| $f_{14}$ | Pressure sensor fault | $ND$ | $ND$ | $ND$ | |
| $f_{15}$ | Positioner feedback fault | | | $DM_X$ | |
| $f_{16}$ | Positioner supply pressure drop | $ND$ | $ND$ | $DM_X$ | |
| $f_{17}$ | Unexpected pressure change across the val. | | | $DM_{F,X}$ | $DM_{F,X}$ |
| $f_{18}$ | Fully or partly opened bypass valves | $DM_F$ | $DM_F$ | $DM_F$ | $DM_F$ |
| $f_{19}$ | Flow rate sensor fault | $DM_F$ | $DM_F$ | $DM_F$ | |

## 5. CONCLUSIONS

The objective of this paper was concerned with modelling and fault detection of an industrial process, especially with obtaining dynamical models directly from the observed data. To tackle this problem, the dynamic GMDH neural network was employed. The main advantage of the presented approach, comparing to the classical MLP, is that GMDH networks have a structure that grows to fit the particular tasks being considered. This enables them to perform better than networks with fixed structures. Another advantage of the proposed technique is that the parameter estimation problem can be formulated as a linear in parameter one. This makes the synthesis of the network fast and effective. Another objective of this work was to provide a description of an industrial application study of the proposed fault detection scheme for a chosen part of the evaporation station at the Lublin sugar factory in Poland. It was shown, using a set of faults, that the proposed GMDH model-based fault detection scheme can provide good results. Experimental results confirm that the most faults under consideration can be detected in a very straightforward way.


## ACKNOWLEDGMENTS

The authors acknowledge funding support under the EC RTN contract (RTN-1999-00392) DAMADICS. Thanks are expressed to the management and staff of the Lublin sugar factory, Cukrownia Lublin SA, Poland for their collaboration and provision of manpower and access to their sugar plant, and in part by the State Committee for Scientific Research (KBN) in Poland.


## 6. REFERENCES


DAMADICS, 2002, Website of the RTN *DAMADICS*. `http://diag.mchtr.pw.edu.pl/damadics`.

Ivakhnenko, A.G. and Muller, J.A., 1995, Self-organizing of nets of active neurons. *System Analysis Modelling Simulation*, **20**, pp. 93-106.

Korbicz, J., Mrugalski, M., and Parisini, T., 2002, Designing state-space models with neural networks. *Proc. 15th IFAC World Congress on Automatic Control*, Barcelona, Spain, 21-26 July, CD-ROM.

Korbicz, J. Patan, K., and Obuchowicz, A., 1999, Dynamic neural networks for process modelling in fault detection and isolation systems. *Int. Appl. Math. and Comp. Sci.*, **9**(3), pp. 519-546.

Mueller, J.E. and Lemke, F., 2000, *Self-organising Data Maining* (Hamburg: Libri).

Mrugalski, M., Arinton, E., and Korbicz, J., 2002, Methods of neuron selection in synthesis of GMDH neural networks. *Proc. 14th Polish Control Conference*, Zielona Góra, Poland, 24-27 June, pp. 845-850.

Nelles, O., 2001, *Non-linear Systems Identification. From Classical Approaches to Neural Networks and Fuzzy Models* (Berlin: Springer).

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