

ROBUST FAULT DETECTION VIA GMDH NEURAL NETWORKS

Marcin Mrugalski * Józef Korbicz * Ron J. Patton **

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

** *Control and Intelligent Systems Engineering,
Department of Engineering, The University of Hull,
Cottingham Road, East Yorkshire HU6 7RX, United Kingdom
e-mail: R.J.Patton@hull.ac.uk*

Abstract: This paper presents a new parameter and confidence estimation techniques for dynamic Group Method of Data Handling Neural Networks (GMDHNNs). The main objective is to show how to employ the bounded error approach to solve such a challenging task that occurs in many practical situations. In particular, the proposed approach can be easily applied in robust fault detection schemes. *Copyright ©2005 IFAC*

Keywords: Identification, neural networks, bounded-error approach, robust fault detection

1. INTRODUCTION

The reliability demands of modern industrial systems require the development of reliable fault diagnosis approaches. During the last few decades many investigations have been made using active approaches, based on residual generation requiring the analytical models (Chen and Patton, 1999; Korbicz *et al.*, 2004; Patton and Korbicz, 1999). Such models can be difficult to obtain for the contemporary complex industrial systems. Furthermore, in the case of the model-based fault diagnosis, the model uncertainty is the elementary factor influencing on the reliability and performance in diagnosing faults. Model uncertainty, as well as disturbances are inevitable in industrial systems, and hence there exists a significant pressure, creating the challenge of robustness in fault diagnosis systems. This paper focuses on the problem of designing Group Method of Data Handling Neural Networks (Mueller and Lemke, 2000) as well as describing their uncertainty (Mrugalski, 2004; Witczak *et al.*, 2005). Knowing the model structure and possessing the knowledge regarding its uncertainty it is possible to design a robust fault detection scheme.

The paper is organized as follows. Section 2 presents the synthesis of the GMDHNN. Section 3 describes selection methods, which can be apply during synthesis. In particular, a method based on the soft selection is presented. Section 4 present sources of GMDHNN uncertainty, whilst section 5 deal with the problem of parameter estimation and outlines some information regarding the so-called bounded-error approach (BEA) for parameter estimation (Milanese *et al.*, 1996). The final part of this work contains an illustrative example, which confirms the effectiveness of the proposed approach.

2. SYNTHESIS OF THE GMDHNN

The concept of the synthesis of the GMDHNN is based on the iterative processing of a defined sequence of operations leading to the evolution of the resulting structure with the application of the appropriate selection methods (as illustrated in Fig. 1), which generates the best approximation of the real system output. The process is completed when the optimal degree of network complexity is achieved. It is assumed that at

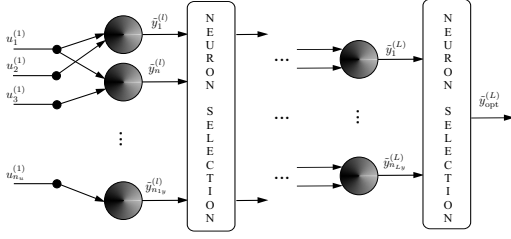


Fig. 1. Synthesis of the GMDH neural network

least two input signals $u_1^{(l)}(k), \dots, u_{n_u}^{(l)}(k)$ constitute the stimulation which results in the formation of the neuron output signal $\tilde{y}_n^{(l)}(k)$:

$$\tilde{y}_n^{(l)}(k) = f(\mathbf{u}) = f(u_1^{(l)}(k), \dots, u_{n_u}^{(l)}(k)), \quad (1)$$

where $\tilde{y}_n^{(l)}(k)$ stands for the neuron output (l is the layer number, n is the neuron number in the l -th layer), corresponding to the k -th measurement of the input $\mathbf{u}(k) \in \mathbb{R}^{n_u}$ of the system. Each neuron in the GMDH network constitutes an elementary model. The parameters of each neuron are estimated separately in such a way that their output signals are the best approximation of the real system 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 (Mrugalski *et al.*, 2003). 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. 2) consists of two submodules: the filter module and the activation module.

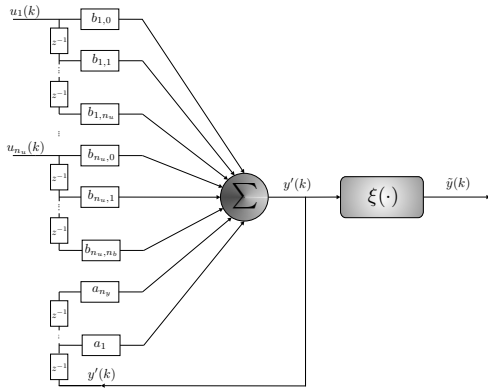


Fig. 2. A dynamic neuron model

The behaviour of the filter module is described by the following equation:

$$\begin{aligned} y_n'^{(l)}(k) = & -a_1 y_n'^{(l)}(k-1) - \dots - a_{n_a} y_n'^{(l)}(k-n_a) \\ & + \mathbf{v}_0^T \mathbf{u}_n^{(l)}(k) + \mathbf{v}_1^T \mathbf{u}_n^{(l)}(k-1) + \dots, \quad (2) \\ & + \mathbf{v}_{n_b}^T \mathbf{u}_n^{(l)}(k-n_b), \end{aligned}$$

or, equivalently,

$$y_n'^{(l)}(k) = \left(\mathbf{r}_n^{(l)}(k) \right)^T \mathbf{p}_n^{(l)}. \quad (3)$$

where $[-y_n'^{(l)}(k-1), \dots, -y_n'^{(l)}(k-n_a), \mathbf{u}_n^{(l)}(k), \mathbf{u}_n^{(l)}(k-1), \dots, \mathbf{u}_n^{(l)}(k-n_b)]$ and $\hat{\mathbf{p}}_n^{(l)} = [a_1, \dots, a_{n_a}, \mathbf{v}_0,$

$\mathbf{v}_1, \dots, \mathbf{v}_{n_b}]$ are the regressor and the filter parameters, respectively. The filter output is used as the input for the activation module

$$\tilde{y}_n^{(l)}(k) = \xi(y_n'^{(l)}(k)). \quad (4)$$

The feature of the above algorithm is that the techniques for the parameter estimation of linear-in-parameter models can be used. Indeed, since $\xi(\cdot)$ is invertible, the neuron described by (2)-(4) can relatively easily be transformed into a linear-in-parameter one. The number of neurons $\tilde{y}_{n_{1y}}^{(l)}$ in the first layer of the network depends on the number of external inputs n_u . In a general case, a network of n_u inputs is built from neurons that have n_p inputs ($n_u > n_p$). In this case, $n_y^{(l)}$ new elements are formed:

$$n_y^{(l)} = \binom{n_y^{(l-1)}}{n_p} = \frac{n_y^{(l-1)}!}{n_p!(n_y^{(l-1)} - n_p)!}. \quad (5)$$

The definition of the evaluation criterion $Q(\hat{y}_n^{(l)})$ of the neurons is a preliminary task in designing a GMDH approach (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 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:

$$\begin{cases} u_1^{(l+1)} = \tilde{y}_1^{(l)}, \\ u_2^{(l+1)} = \tilde{y}_2^{(l)}, \\ \dots \\ u_{n_{1y}}^{(l+1)} = \tilde{y}_{n_{1y}}^{(l)}. \end{cases} \quad (6)$$

In analogous way, the new neurons in the next layers of the network are created. During the synthesis of the GMDHNN, the number of layers suitably increases. Each time when a new layer is added, new neurons are introduced. The synthesis of the GMDHNN is completed when the network fits the data with desired accuracy or the introduction of new neurons did not induce a significant increase in the approximation abilities of the neural network. In order to achieve this goal it is necessary to calculate the quality index $Q(\hat{y}_n^{(l)})$ for all n_y neurons included in the l layer. The $Q_{\min}^{(l)}$ represents the processing error for the best neuron in this layer

$$Q_{\min}^{(l)} = \min_{n=1, \dots, n_y} Q(\hat{y}_n^{(l)}). \quad (7)$$

The values $Q(\hat{y}_n^{(l)})$ 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 synthesis of the GMDHNN is completed when the following condition occurs:

$$Q_{\text{opt}}^{(L)} = \min_{l=1, \dots, L} Q_{\min}^{(l)}. \quad (8)$$

The $Q_{\text{opt}}^{(L)}$ 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, all unnecessary neurons are removed, leaving only those which are relevant to the computation of the model output. The procedure of removing unnecessary neurons is the last stage of the synthesis of the GMDHNN.

3. SELECTION METHODS IN THE GMDHNN

The selection methods in the GMDHNN plays a role of a mechanism of the structural optimization at the stage of construing a new layer of neurons. Only well performing neurons, which outputs are the best approximation of the system output signal, are preserved to build a new layer. The output of the neuron may becomes an input to other neurons in the next layer or an output of the model. During the selection, neurons which have too large defined quality index $Q(\hat{y}_n^{(l)})$ are rejected based on chosen selection methods. There exist a few methods of performing the selection procedure. One of the most often applied is a *constant population method* (Mueller and Lemke, 2000), which is based on selection of g neurons, for which an evaluation criterion $Q(\hat{y}_n^{(l)})$ reaches the least values. The constant g is chosen on empirical way. The most important advantage of this method is the simplicity of implementation. Unfortunately, constant population method has very restrictive structure evolution possibilities. The similar situation is in the case of application of the *decreasing population method*. This method defines the maximum number of elements in layer. The number of the neurons in each layer decreases along with the growth of the network. One way out of this problem is an application of the *optimal population method*. This approach is based on rejecting the neurons for which the defined quality index is larger than an arbitrarily determined threshold e_h . Usually, threshold is determined separately for each layer and depends on the quality index for a current layer. The threshold is selected in an empirical way and is dependent of the considered task. Difficulty with the selection of the threshold cause that the optimal population method is not applied too often. One of ways of performing the selection procedure is an application of the method based on the soft selection approach. Thanks to a proper choice of the quantity signals to the selection procedure, the method achieve property of the soft selection. The *soft selection method* (Mrugalski, 2004) is divided into three parts as shown in Table 1. The property of the soft selection follows from the specific series of competitions. It may happen, that the potentially unfitted neuron will be selected. Everything depends on its score in the series of competition.

Table 1. The soft selection method

Input : The set of all n_y neurons in the l -th layer, n_j - the number of a opponent neurons, n_w - the number of winnings required for the n -th neuron selection.

Output : The set of neurons after selection.

- (1) Calculate the evaluation criterion $Q(\hat{y}_n^{(l)})$ for $n = 1, \dots, n_y$ neurons
- (2) Conduct series of n_y competitions between each n -th neuron in the layer and a n_j randomly selected neurons (the so-called *opponent*) from the same layer. The n -th neuron is so-called *winner* neuron when:

$$Q(\hat{y}_n^{(l)}) \leq Q(\hat{y}_j^{(l)}), \quad j = 1, \dots, n_j$$

where $\hat{y}_j^{(l)}$ denotes a signal generated by the opponent neuron

- (3) Selection of the neurons for the $l + 1$ -th layer with the number of winnings bigger then n_w (the remaining neurons are removed)
-

In this way, distinct from other selection methods, it is possible to use potentially unfitted neurons which in the next layers may improve the quality of the model. Moreover, if the neural network is not fitted perfectly to the identification data set, it is possible to achieve a network which possess better generalization abilities. One of the most important parameters which should be chosen in the selection proces is the number of the opponents n_j . The bigger value of n_j makes that the probability of the selection of a neuron with little quality index is low. In this way, in extreme situation when $n_j \gg n_y$ the soft selection method will behave like the constant population method which is based on the selection only of the best fitted neurons. Some experimental results performed on a number of selected examples indicate that soft selection method makes it possible to obtain a more flexible network structure. Another advantage, comparing to the optimal population method, is that we avoid an arbitrary selection of the threshold. Instead of this we have to select a number of winnings n_w . This is, of course, a less sophisticated task.

4. UNCERTAINTY OF THE GMDH MODEL

In order to perform the model construction procedure it is necessary to define the quality index. Mueller and Lemke (2000) present a comprehensive table of the most common quality indexes used in the parametric GMDH algorithm. The most often applied are *Akaike Information Criterion* (AIC) and *Final Prediction Error* (FPE). These criterions are based on the statistic taking into consideration the complexity of elementary models. The optimal structure of the elementary model is obtained when the statistic has the minimal value. In the case of AIC criterion statistic has the following general form:

$$W_{n_D} = n_D \log J_{n_D}(N_{\text{arch}}) + \gamma(n_D, n_p), \quad (9)$$

where $J_{n_{\mathcal{D}}}(N_{\text{arch}})$ represent the goal function for the model architecture N_{arch} and $\gamma(n_{\mathcal{D}}, n_p)$ is the function of the number of the data samples $n_{\mathcal{D}}$ and the number of elementary model parameters n_p . The appropriate selection of the (9) ensure its increasing along with increasing of the number of parameters and converge to zero along with increasing of the data samples set. The selection of the function characterized by the above mentioned properties ensure an elimination of the over-parameterized elementary models. In the case of the AIC criterion the factor $\gamma(n_{\mathcal{D}}, n_p)$ is equal $2n_p$ what lead to the following final form of the criterion:

$$W_{n_{\mathcal{D}}} = n_{\mathcal{D}} \log J_{n_{\mathcal{D}}}(N_{\text{arch}}) + 2n_p. \quad (10)$$

In the case of the FPE criterion the statistic reflect an expected variance of prediction error during prediction new observations based on the model obtained for the identification data set.

$$W_{n_{\mathcal{D}}} = \mathcal{E}(s_{\tau}^2(\tau, N_{\text{arch}})), \quad (11)$$

where τ denote the prediction period. In (Soderstrom and Stoica, 1989) was shown, that the statistic (11) can be approximate by the following expression:

$$W_{n_{\mathcal{D}}} \approx \Lambda(1 + n_p/n_{\mathcal{D}}), \quad (12)$$

where an asymptotic unbiased estimate of the Λ is:

$$\hat{\Lambda} = \frac{J_{n_{\mathcal{D}}}(N_{\text{arch}})}{(1 - n_p/n_{\mathcal{D}})}. \quad (13)$$

As a result of substituting (13) into (12) the final form of the FPE criterion is obtained:

$$W_{n_{\mathcal{D}}} = J_{n_{\mathcal{D}}}(N_{\text{arch}}) \frac{1 + n_p/n_{\mathcal{D}}}{1 - n_p/n_{\mathcal{D}}}. \quad (14)$$

In the case of AIC criterion, it is possible to select better elementary model based on the inequality defined with the statistic (9):

$$\begin{aligned} n_{\mathcal{D}} \log J_{n_{\mathcal{D}}}(N_{\text{arch},1}) + \gamma(n_{\mathcal{D}}, n_{p,1}) &\leq \\ n_{\mathcal{D}} \log J_{n_{\mathcal{D}}}(N_{\text{arch},2}) + \gamma(n_{\mathcal{D}}, n_{p,2}) & \end{aligned} \quad (15)$$

where after simple transformation has a form:

$$\begin{aligned} J_{n_{\mathcal{D}}}(N_{\text{arch},1}) &\leq J_{n_{\mathcal{D}}}(N_{\text{arch},2}) \cdot \\ \exp \left[\frac{(\gamma(n_{\mathcal{D}}, n_{p,2}) - \gamma(n_{\mathcal{D}}, n_{p,1}))}{n_{\mathcal{D}}} \right] & \end{aligned} \quad (16)$$

and finally:

$$\begin{aligned} \chi_{\alpha}^2(n_{p,2} - n_{p,1}) &= n_{\mathcal{D}} \cdot \\ \left(\exp \left[\frac{(\gamma(n_{\mathcal{D}}, n_{p,2}) - \gamma(n_{\mathcal{D}}, n_{p,1}))}{n_{\mathcal{D}}} \right] - 1 \right) & \end{aligned} \quad (17)$$

Based on the (17) the AIC criterion can be perceived as the F-test (Soderstrom and Stoica, 1989) with in advance defined confidence level. The same disadvantage occurs in the case of the FPE criterion. In (Soderstrom and Stoica, 1989) was theoretically and practically proved, that for $n_{p,2} - n_{p,1} = 1$ degree of freedom, the confidence level is 0.157. This result means, that the probability of selection over-parameterized structure $N_{\text{arch},2}$ via AIC or FPE criterions is 15.7%. If the number of elementary models

in the GMDHNN is high then probability of selection over-parameterized neurons is not acceptable. Another reason opposite the application of the AIC and FPE criterions is fact, that the probability of selection over-parameterized elementary neurons is not decreasing along with $n_{\mathcal{D}} \rightarrow \infty$. Furthermore, the AIC and FPE criterions were designed for comparison of the hierarchical elementary models $N_{\text{arch},1} \subset N_{\text{arch},2}$. In the case of the GMDHNN this assumption is not fulfilled (Fig. 3). Apart from the model structure selec-

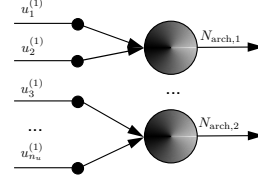


Fig. 3. The problem of unhierarchy of the neurons in the first layer of the GMDHNN

tion stage, inaccuracy in parameter estimates also contributes to modelling uncertainty (Mrugalski, 2004).

5. CONFIDENCE ESTIMATION OF GMDHNN

Let us consider the following system:

$$y(k) = \left(\mathbf{r}_n^{(l)}(k) \right)^T \mathbf{p}_n^{(l)} + \varepsilon_n^{(l)}(k). \quad (18)$$

The problem is to obtain the parameter estimate vector $\hat{\mathbf{p}}_n^{(l)}(k)$, as well as an associated parameter uncertainty required to design robust fault detection system (Mrugalski, 2004; Witczak *et al.*, 2005). In order to simplify the notation, the index $n^{(l)}$ is omitted. The knowledge regarding the set of admissible parameter values allows obtaining the confidence region of the model output which satisfies

$$\tilde{y}^m(k) \leq y(k) \leq \tilde{y}^M(k), \quad (19)$$

where $y^m(k)$ and $y^M(k)$ are the minimum and maximum admissible values of the model output that are consistent with the input-output measurements of the system. In this paper, it is assumed that $\varepsilon(k)$ consists of a structural deterministic error caused by the model-reality mismatch, and the stochastic error caused by the measurement noise is bounded as follows

$$\varepsilon^m(k) \leq \varepsilon(k) \leq \varepsilon^M(k), \quad (20)$$

where the bounds $\varepsilon^m(k)$ and $\varepsilon^M(k)$ ($\varepsilon^m(k) \neq \varepsilon^M(k)$) can be estimated (Witczak *et al.*, 2005). The idea underlying the bounded-error approach is to obtain a feasible parameter set (Milanese *et al.*, 1996). This set can be defined as

$$\mathbb{P} = \{ \mathbf{p} \in \mathbb{R}^{n_p} \mid y(k) - \varepsilon^M(k) \leq \mathbf{r}^T(k) \mathbf{p} \leq y(k) - \varepsilon^m(k), \quad k = 1, \dots, n_{\mathcal{T}} \}, \quad (21)$$

where $n_{\mathcal{T}}$ is the number of input-output measurements. This set can be perceived as a region of parameter space that is determined by $n_{\mathcal{T}}$ pairs of hyperplanes where each pair defines the parameter strip:

$$\mathbb{S}(k) = \{ \mathbf{p} \in \mathbb{R}^{n_p} \mid y(k) - \varepsilon^M(k) \leq \mathbf{r}^T(k) \mathbf{p} \leq y(k) - \varepsilon^m(k) \}, \quad (22)$$

and hence

$$\mathbb{P} = \bigcap_k^{n_T} \mathbb{S}(k). \quad (23)$$

Let \mathbb{V} be the set of all vertices \mathbf{p}^i , $i = 1, \dots, n_v$, describing the feasible parameter set \mathbb{P} . If there is no error in the regressor, then the problem of determining the model output uncertainty can be solved as follows:

$$\mathbf{r}^T(k)\mathbf{p}^m(k) \leq \mathbf{r}^T(k)\mathbf{p} \leq \mathbf{r}^T(k)\mathbf{p}^M(k), \quad (24)$$

where

$$\mathbf{p}^m(k) = \arg \min_{\mathbf{p} \in \mathbb{V}} \mathbf{r}^T(k)\mathbf{p}, \quad (25)$$

$$\mathbf{p}^M(k) = \arg \max_{\mathbf{p} \in \mathbb{V}} \mathbf{r}^T(k)\mathbf{p}. \quad (26)$$

As is has already been mentioned, the neurons in the l -th ($l > 1$) layer are fed with the outputs of the neurons from the $(l - 1)$ -th layer. In order to modify the above presented approach for the uncertain regressor case, let us denote an unknown ‘‘true’’ value of the regressor $\mathbf{r}_n(k)$ by a difference between a known (measured) value of the regressor $\mathbf{r}(k)$ and the error in the regressor $\mathbf{e}(k)$:

$$\mathbf{r}_n(k) = \mathbf{r}(k) - \mathbf{e}(k), \quad (27)$$

where it is assumed that the error $\mathbf{e}(k)$ is bounded as:

$$\varepsilon_i^m(k) \leq e_i(k) \leq \varepsilon_i^M(k), \quad i = 1, \dots, n_p. \quad (28)$$

Using (18) and substituting (27) into (28) one can define the space containing the parameter estimates:

$$\begin{aligned} \varepsilon^m(k) - \mathbf{e}^T(k)\mathbf{p} &\leq y(k) - \mathbf{r}(k)^T\mathbf{p} \leq \\ \varepsilon^M(k) - \mathbf{e}^T(k)\mathbf{p}. \end{aligned} \quad (29)$$

Unfortunately, for the purpose of parameter estimation it is not enough to introduce (27) into (28). Indeed, the bounds of (29) depend also on the sign of each p_i and these signs are in general unknown. The best way out of this problem, is to replace them by

$$p_i = p'_i - p''_i, \quad p'_i, p''_i \geq 0, \quad i = 1, \dots, n_p. \quad (30)$$

Although the above solution is very simple, it doubles the number of parameters, i.e. instead of estimating n_p parameters it is necessary to do so for $2n_p$ parameters. In spite of that, this technique is very popular and widely used in the literature (Milanese *et al.*, 1996). Due to the above solution, (29) can be modified as follows:

$$\begin{aligned} \varepsilon^m(k) - (\mathbf{e}^M(k))^T\mathbf{p}' + (\mathbf{e}^m(k))^T\mathbf{p}'' \\ \leq y(k) - \mathbf{r}^T(k)(\mathbf{p}' - \mathbf{p}'') \leq \\ \varepsilon^M(k) - (\mathbf{e}^m(k))^T\mathbf{p}' + (\mathbf{e}^M(k))^T\mathbf{p}''. \end{aligned} \quad (31)$$

The proposed modification of the BEA makes it possible to estimate the parameter vectors of the neurons from the l -th, $l > 1$ layers. In the case of an error in the regressor, using (31), it can be shown that the model output uncertainty has the following form:

$$\begin{aligned} \tilde{y}^m(k)(\mathbf{p}'^m(k), \mathbf{p}''^m(k)) \leq \mathbf{r}_n^T\mathbf{p} \leq \\ \tilde{y}^M(k)(\mathbf{p}'^M(k), \mathbf{p}''^M(k)), \end{aligned} \quad (32)$$

where

$$\begin{aligned} \tilde{y}^m(k)(\mathbf{p}'^m(k), \mathbf{p}''^m(k)) = (\mathbf{r}(k) - \mathbf{e}^M(k))^T \\ \mathbf{p}'^m(k) + (\mathbf{e}^m(k) - \mathbf{r}(k))^T\mathbf{p}''^m(k), \end{aligned} \quad (33)$$

$$\begin{aligned} \tilde{y}^M(k)(\mathbf{p}'^M(k), \mathbf{p}''^M(k)) = (\mathbf{r}(k) - \mathbf{e}^m(k))^T \\ \mathbf{p}'^M(k) + (\mathbf{e}^M(k) - \mathbf{r}(k))^T\mathbf{p}''^M(k), \end{aligned} \quad (34)$$

and

$$(\mathbf{p}'^m(k), \mathbf{p}''^m(k)) = \arg \min_{(\mathbf{p}', \mathbf{p}'') \in \mathbb{V}} \tilde{y}^m(k)(\mathbf{p}', \mathbf{p}''(k)), \quad (35)$$

$$(\mathbf{p}'^M(k), \mathbf{p}''^M(k)) = \arg \max_{(\mathbf{p}', \mathbf{p}'') \in \mathbb{V}} \tilde{y}^M(k)(\mathbf{p}', \mathbf{p}''(k)). \quad (36)$$

Using (32) it is possible to obtain the system output uncertainty:

$$\begin{aligned} \tilde{y}^m(k)(\mathbf{p}'^m(k), \mathbf{p}''^m(k)) + \varepsilon^m(k) \leq y(k) \leq \\ \tilde{y}^M(k)(\mathbf{p}'^M(k), \mathbf{p}''^M(k)) + \varepsilon^M(k). \end{aligned} \quad (37)$$

In order to adapt of the presented approach to the parameter estimation of non-linear neurons, it is necessary to transform the relation

$$\varepsilon^m(k) \leq y'(k) - \xi((\mathbf{r}(k))^T\mathbf{p}) \leq \varepsilon^M(k) \quad (38)$$

using $\xi^{-1}(\cdot)$, and hence

$$\begin{aligned} \xi^{-1}(y(k) - \varepsilon^M(k)) \leq (\mathbf{r}(k))^T\mathbf{p} \leq \\ \xi^{-1}(y(k) - \varepsilon^m(k)). \end{aligned} \quad (39)$$

As has been already pointed out, an error in the regressor must be taken into account during the design procedure of the neurons from the second and the subsequent layers. Indeed, by using (24) in the first layer and (32) in the subsequent ones, it is possible to obtain the bounds of the output (3) and the bounds of the regressor error (20). Note that the processing errors of the neurons, which are described by the model output uncertainty (32), can be propagated and accumulated during the introduction of new layers. This unfavourable phenomenon can be reduced by the application of soft selection method. Unfortunately, as has already been mentioned in section 4, the application of the classical evaluation criteria during the network synthesis may lead to the selection of an inappropriate structure of the GMDHNN. This follows from the fact that the above criteria do not take into account the modelling uncertainty. In this way, neurons with small values of classical quality indexes but with large uncertainty can be obtained. In order to overcome this difficulty, a new evaluation criterion of the neurons has been introduced in this work, i.e.

$$Q_V = \frac{1}{n_V} \sum_{k=1}^{n_V} |(\tilde{y}^M(k) + \varepsilon^M(k)) - (\tilde{y}^m(k) + \varepsilon^m(k))| \quad (40)$$

where n_V is the number of input-output measurements for the validation data set, $\tilde{y}^M(k)$ and $\tilde{y}^m(k)$ are calculated with (24) for the first layer or with (33)-(34) for the subsequent ones. Finally, the neuron in the last layer that gives the smallest processing error (40) constitutes the output of the GMDHNN and the system output uncertainty interval for this neuron can be used for robust fault detection.

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 data from GARTEUR benchmark were employed to identify the input-output model of the low-fidelity Boeing 747-100/200 aircraft model (Esteban and Balas, 2003). The main difference between high- and low-fidelity models is a reduction of the stability derivatives in the aerodynamic coefficients. In order to obtain the training data the aircraft was trimmed at an equilibrium point. The selected aircraft mass was 300,000 kg, and the position of the aircraft's center of gravity with to the (x, y, z) -axes was assumed to be 25 percent of the mean aerodynamic chord for the x -axis and the point (0,0) meters for the other two axes. No fault are assumed, and a flight condition defined to be straight-level-flight at 7000 meters of altitude and at a true airspeed of 241 m/sec was given. During flight simulation the following pilot inputs were used: $stab$ – stabilizer, δ_w – wheel, δ_p – pedal and δ_c – column. Table 2 gives the low fidelity longitudinal and lateral aircraft states. The data used for the identification

Table 2. Aircraft states

q_{body}	Pitch rate	p_{body}	Roll rate
$VTAS$	True Air Speed	r_{body}	Yaw rate
α	Angle of attack	β	Sideslip angle
θ	Pitch angle	ϕ	Roll angle
he	Altitude	ψ	Yaw angle
xe	x-position	ye	y-position

set were 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 selection of best performing neurons for their processing accuracy is realized with application of the soft selection method based on the proposed evaluation criterion (40). For the fault detection purpose fault scenario containing wing damage due to engine separation was simulated. The Fig. 4 present the real system response (yaw rate state) as well as the corresponding system output uncertainty obtained with the GMDH approach for this scenario. An occurrence of fault is signalled by the violation of the system output uncertainty interval by the real system response. As can be seen the fault is very easy to detect.

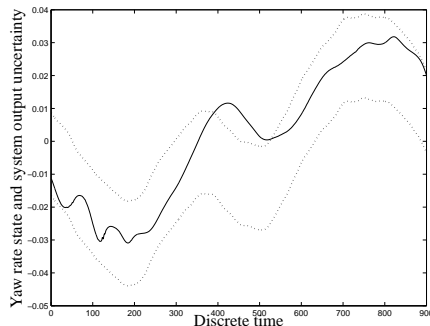


Fig. 4. The real yaw rate state as well as the corresponding system output uncertainty

7. CONCLUSIONS

The objective of this paper was concerned with obtaining models and calculating their uncertainty directly from the observed data. It was shown how to estimate parameters and the corresponding uncertainty of an individual elementary model and the whole GMDH neural network. Based on the GMDH neural network, a novel robust fault detection scheme was proposed which supports the diagnostic decisions. The proposed approach for system identification and fault detection was tested on the GARTEUR benchmark problem.

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.

REFERENCES

- Chen, J. and R. J. Patton (1999). *Robust Model-Based Fault Diagnosis For Dynamic Systems*. Kluwer Academic Publ, London.
- Esteban, A. M. and G.J. Balas (2003). *A Boeing 747-100/200 Aircraft Fault Tolerant and Fault Diagnostic Benchmark*. Aerospace Engineering and Mechanics Department, University of Minnesota, Minnesota, technical report.
- Korbicz, J., Kościelny, J.M., Kowalczyk, Z. and W. Cholewa (Eds.) (2004). *Fault Diagnosis. Models, Artificial Intelligence, Applications*. Springer, Berlin.
- Milanese, M., Norton, J., Piet-Lahanier, H. and E. Walter (Eds.) (1996). *Bounding Approaches to System Identification*. Plenum Press, New York.
- Mrugalski, M., Arinton, E. and J. Korbicz (2003). *Fault detection with dynamic GMDH neural networks: application to the DAMADICS benchmark problem*. – Proc. 5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS 2003, Washington, USA, pp.1071–1076.
- Mrugalski, M. (2004). *Neural Network Based Modelling of Non-linear Systems in Fault Detection Schemes*. Faculty of Electrical Engineering, Computer Science and Telecommunications, University of Zielona Góra, Zielona Góra, doctoral dissertation, (In Polish).
- Mueller, J.E. and F. Lemke (2000). *Self-organising Data Maining*. Libri, Hamburg.
- Patton, R.J. and J. Korbicz (Eds.) (1999). Advances in computational intelligence for fault diagnosis systems. *Applied Mathematical and Computer Science*, special issue, 9(3), pp.468–735.
- Soderstrom, T. and P. Stoica (1989). *System Identification*. Prentice-Hall Inter., Hemel Hempstead.
- Witczak, M., Korbicz, J., Mrugalski, M. and R.J. Patton (2005). A GMDH neural network based approach to robust fault detection and its application to solve the DAMADICS benchmark problem. *Control Engineering Practice*, (accepted).