amcs

AN UNSCENTED KALMAN FILTER IN DESIGNING DYNAMIC GMDH NEURAL NETWORKS FOR ROBUST FAULT DETECTION

MARCIN MRUGALSKI

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

This paper presents an identification method of dynamic systems based on a group method of data handling approach. In particular, a new structure of the dynamic multi-input multi-output neuron in a state-space representation is proposed. Moreover, a new training algorithm of the neural network based on the unscented Kalman filter is presented. The final part of the work contains an illustrative example regarding the application of the proposed approach to robust fault detection of a tunnel furnace.

Keywords: robust fault detection, non-linear system identification, dynamic GMDH neural network, unscented Kalman filter.

1. Introduction

The effectiveness of contemporary *Fault Detection and Isolation* (FDI) systems (Ding, 2008; Korbicz *et al.*, 2004; Korbicz and Kościelny, 2010; Palade *et al.*, 2006; Witczak, 2007) and *Fault Tolerant Control* (FTC) schemes (Ichalal *et al.*, 2012; Isermann, 2005; Niemann, 2012; Noura *et al.*, 2009; Montes de Oca *et al.*, 2012) mostly depends on the quality of the models obtained in the process of system identification. *Artificial Neural Networks* (ANNs) are often applied in the process of dynamic non-linear industrial system identification (Gupta *et al.*, 2003; Haykin, 2009; Mrugalski *et al.*, 2008). Their attractiveness follows from the fact that in the case of many industrial systems their models created on the basis of the physical laws describing the system behavior (Isermann, 2005; Korbicz *et al.*, 2004; Patton *et al.*, 2000) are too complex or often unavailable. Additional advantages of ANNs are properties such as generalization abilities, the ability to learn and good approximation of non-linear systems.

Unfortunately, ANNs have also some important disadvantages, which limit the effectiveness of the developed FDI and FTC systems. The most important disadvantages are inefficient quality of the neural model, immature approaches allowing the modeling of dynamics, and usually an unavailable description of the neural model in a state-space representation. Also only few approaches ensure the stability of neural models during the process of dynamic system identification. Moreover, there is a limited number of approaches that allow a mathematical description of neural model uncertainty and this factor has major impact on the effectiveness of FDI and FTC systems (Witczak, 2007; Patan *et al.*, 2008).

One of the most important problems of neural modeling is the appropriate selection of the neural model structure. The errors following from an inappropriate selection of the neural network architecture and those related to inaccurate estimation of the parameters have deciding impact on the quality of the neural model resulting in the occurrence of model uncertainty (Korbicz and Mrugalski, 2008; Mrugalski and Korbicz, 2007; Patan *et al.*, 2008). To tackle this problem, the *Group Method of Data Handling* (GMDH) approach can be employed (Ivakhnenko and Mueller, 1995; Korbicz and Mrugalski, 2008; Mrugalski and Witczak, 2012). The concept of the GMDH approach relies on replacing the complex neural model by a set of hierarchically connected partial models, which can be chosen with appropriate selection methods. The proposed approach also allows developing the formula of the GMDH model due to the inclusion of additional procedures, which can be used to extend the scope of the application. The GMDH model can be used in the identification of static and dynamic systems, both *Single-Input Single-Output* (SISO) and *Multi-Input Multi-Output* ones (MIMO) (Mrugalski and

Korbicz, 2011; Korbicz and Mrugalski, 2008; Mrugalski *et al.*, 2003).

Another important problem of neural modeling is concerned with the identification of dynamic systems. In the case of the classical neural network, for example, the *Multi-Layer Perceptron* (MLP), the modelling problem of the dynamics is tried to be solved by the introduction of additional, suitably delayed global output feedback lines. Moreover, several fully or partially recurrent neural networks were developed, i.e., the Real Time Recurrent Network (RTRN) network (Williams and Zipser, 1989), Elman's network (Elman, 1990) or Jordan's network (Jordan and Bishop, 1997). Unfortunately, the process of training these networks and ensuring their stability is quite difficult. On the other hand, locally recurrent but globally feed-forward networks with dynamic neurons can be applied to the identification of the dynamic systems. Due to the introduction of diverse local feedback to the classical neuron model, it is possible to achieve a few types of dynamic neurons, e.g., with local activation feedback (Fasconi *et al.*, 1992), local synapse feedback (Back and Tsoi, 1991) and output feedback (Gori *et al.*, 1989). The main advantage of networks constructed with the application of dynamic neurons is that their stability can be proved relatively easily. As a matter of the fact, the stability of the network only depends on that of neurons. The feed-forward structure of such networks seems to make the training process easier. On the other hand, the introduction of dynamic neurons increases the parameter space significantly. This drawback, together with the non-linear and multi-modal properties of dynamic neurons, implies that parameter estimation becomes relatively complex. Moreover, all the above mentioned dynamic neural models have no state-space description. In fact, approaches trying to solve such a challenging problem can be rarely found in the literature (Pan *et al.*, 2001; Zamarreño and Vega, 1998). Unfortunately, these approaches do not allow calculating the uncertainty of these models, which is necessary to apply them in robust fault detection schemes.

To tackle all the above mentioned problems, the GMDH neural network can be applied. The behaviour of each partial model in the GMDH neural network should reflect that of the identified system. It follows from the rule of the GMDH algorithm that the parameters of each partial model are estimated in such a way that their output is the best approximation of the real system output. In this situation, the partial model should have an ability to represent the dynamics. To solve such a problem, in this paper a new structure of the multi-input and multi-output dynamic neuron in the state-space representation is proposed.

The description of the dynamic neuron in the suitable state-space representation enables to obtain
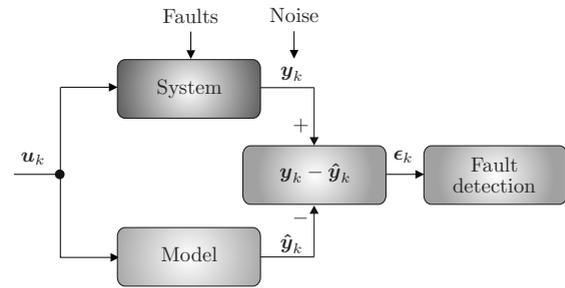
Fig. 1. Scheme of the residual-based fault detection system.

constraints of parameter estimates which ensure the stability of each dynamic neuron and, as a consequence, the stability of the whole GMDH network as well. The description of dynamic neurons in the state-space representation, together with the fact that in the GMDH model the parameters of each neuron are estimated separately, easily allows applying advanced parameters estimation algorithms, e.g., the *Unscented Kalman Filter* (UKF) (Haykin, 2001; Teixeira *et al.*, 2010; Witczak and Prętki, 2007). Thus, UKF-based constrained parameter estimation ensures asymptotically stable neurons of the GMDH model. Another advantage of the UKF is that it allows obtaining a description of neural model uncertainty, which can then be applied to robust fault detection (Ding, 2008; Witczak, 2007).

In this work, the structure of the multi-input multi-output GMDH neural network is portrayed. For such a network, dynamic neurons in the state-space representation are used to represent the dynamics of the system. Subsequently, the application of the UKF to parameter estimation of dynamic neurons is presented. Moreover, the concept of the application of the UKF to robust fault detection is shown. As an example, which confirms the effectiveness of the proposed approach, identification and robust fault detection of a tunnel furnace is considered.

2. Robust fault diagnosis

Model-based fault diagnosis can be defined as the detection, isolation and identification of faults in the system based on a comparison of the available system measurements with information represented by a mathematical model (Fig. 1) (Patton *et al.*, 2000; Korbicz *et al.*, 2004). The model of the diagnosed system is developed before its application in the fault diagnosis system.

The comparison of the system $\boldsymbol{y}_k$ and the model response $\hat{\boldsymbol{y}}_k$ leads to the generation of the residual

$$\boldsymbol{\epsilon}_k = \boldsymbol{y}_k - \hat{\boldsymbol{y}}_k. \qquad (1)$$

In the model-based fault detection approach, it is
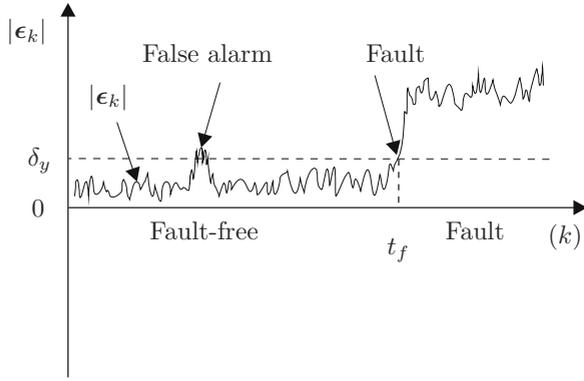
Fig. 2. Residual-based fault detection.
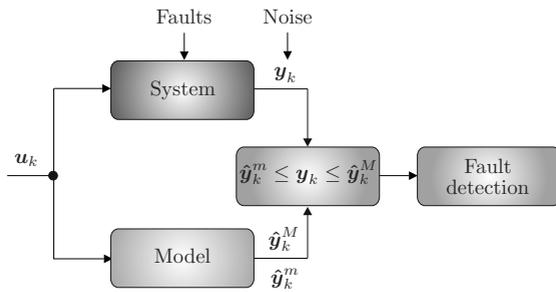
Fig. 3. Scheme of the robust fault detection system.

assumed that the residual ϵ_k should be close to zero in the fault-free case, and it should be distinguishably different from zero in the faulty case. Under such an assumption, faults are detected by the application of a fixed threshold on the residual signal (cf. Fig. 2).

In this case, the fault is signaled when the absolute value of the residual $|\epsilon_k|$ is larger then an arbitrarily assumed threshold δ_y:

$$|\epsilon_k| \le \delta_y. \qquad (2)$$

The difficulty with such a kind of symptom evaluation approach is the corruption of $\boldsymbol{y}_k$ by noise and disturbances. Another difficulty follows from the fact that the model obtained during system identification is usually uncertain (Witczak *et al.*, 2006). Model uncertainty can appear during both stages of system identification, i.e., model structure selection and parameter estimation. To tackle this problem, the adaptive time-variant threshold, which is adapted according to the system behaviour, can be applied. The concept of the proposed approach is illustrated in Fig. 3.

The main idea of this approach relies on the calculation of the system output adaptive threshold according the following equations (Witczak, 2007):

$$\hat{y}_{i,k}^m = \hat{y}_{i,k} - t_{n_t-n_p}^{\alpha/2} \hat{\sigma}_i \left(1 + \boldsymbol{u}_{i,k} \boldsymbol{P} \boldsymbol{u}_{i,k}^T\right)^{1/2}, \qquad (3)$$

$$\hat{y}_{i,k}^M = \hat{y}_{i,k} + t_{n_t-n_p}^{\alpha/2} \hat{\sigma}_i \left(1 + \boldsymbol{u}_{i,k} \boldsymbol{P} \boldsymbol{u}_{i,k}^T\right)^{1/2}, \qquad (4)$$

where $\boldsymbol{u}_{i,k}$ is the regressor vector while $\boldsymbol{P}$ denotes the covariance matrix of the parameters being estimated, $i = 1, \ldots, n_y$ is the number of the model outputs, $t_{n_t-n_p}^{\alpha/2}$ is the corresponding t-Student distribution quantile, n_t is the number of data points in the training data set, n_p is the number of parameters of the neural model and $\hat{\sigma}_i$ is the standard deviation of the i-th fault-free residual. The output interval (Fig. 4) should contain the real system response in the fault-free mode.
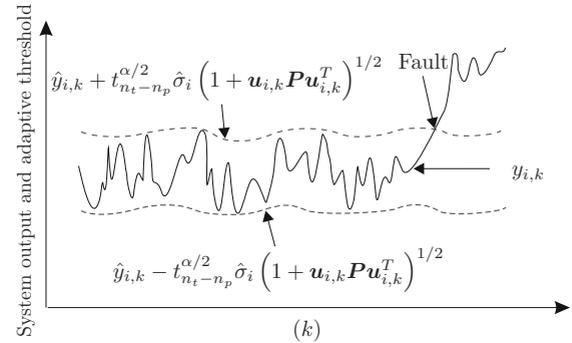
Fig. 4. Application of the system output adaptive threshold to robust fault detection.

3. Synthesis of GMDH neural models

The concept of the GMDH approach relies on replacing the complex neural model by the set of hierarchically connected partial models (neurons). It should be pointed out that the parameters of each newly created neuron are estimated separately. The neurons are evaluated, selected and included in the newly created neuron layers. During network synthesis, new layers of neurons are successively created and added to the GMDH network. The process of network synthesis leads to the evolution of the resulting model structure to obtain the best quality approximation of real system output signals. The process is completed when an optimal degree of network complexity is achieved (Ivakhnenko and Mueller, 1995). An outline of the GMDH algorithm is presented in Fig. 5.

The above assumptions of GMDH networks offer a lot of freedom in defining particular elements of the algorithm. The possibilities relate to, for example, the structure of the partial model, the definition of the transition function, evaluation criteria of the processing accuracy or selection methods.

Let us assume that in a general case each neuron in the GMDH network has the following form:

$$\hat{\boldsymbol{s}}_{i,j,k}^{(l)} = \mathcal{F}\left(\boldsymbol{r}_{i,k}^{(l)}, \boldsymbol{p}_{i,j}^{(l)}\right), \qquad (5)$$
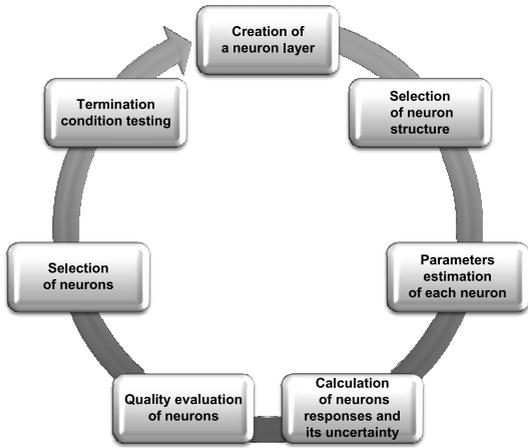
Fig. 5. Synthesis process of the GMDH neural network.

Fig. 6. Design of the first layer of neurons in the GMDH neural network.

where $r_{i,k}^{(l)} \in \mathbb{R}^{n_r}$ for $i = 1, \ldots, n_R$ are the neuron input vectors formed as combinations of the neural model inputs $r_{i,k}^{(l)} = [u_{i,k}^{(l)}, \ldots, u_{j,k}^{(l)}]^T$, $\hat{s}_{i,j,k}^{(l)} \in \mathbb{R}^{n_s}$ for $j = 1, \ldots, n_N$ are the neuron output vectors formed as combinations of the network outputs $[\hat{y}_{i,k}^{(l)}, \ldots, \hat{y}_{j,k}^{(l)}]^T$, $p_{i,j}^{(l)} \in \mathbb{R}^{n_r \times n_s}$ denotes the parameter estimate matrix, $\mathcal{F}(\cdot)$ is an activation function, and l is the number of layers of the GMDH network.

The process of the synthesis of the first layer of the GMDH neural network begins with the creation of a set of n_R vectors of neuron inputs $r_{i,k}^{(l)}$ based on the combinations of the model inputs $u_k \in \mathbb{R}^{n_u}$ belonging to the training data set $\mathcal{T}$. The number of the vectors $r_{i,k}^{(l)}$ depends on that of model inputs n_u and neuron inputs n_r. Each i-th vector $r_{i,k}^{(l)}$ constitutes a neuron stimulation which results in the formation of j-th neurons and their outputs $\hat{s}_{i,j,k}^{(l)}$, which are the estimates of the modeled system outputs. The number n_N of these neurons, for each subsequent i-th vector $r_{i,k}^{(l)}$, depends on the number of the modeled output signals n_y and an assumed number of the neurons inputs n_r:

$$\begin{cases} \hat{s}_{1,1,k}^{(1)} = \mathcal{F}(r_{1,k}^{(1)}), \\ \quad \vdots \\ \hat{s}_{1,n_N,k}^{(1)} = \mathcal{F}(r_{1,k}^{(1)}), \\ \quad \vdots \\ \hat{s}_{n_R,1,k}^{(1)} = \mathcal{F}(r_{n_R,k}^{(1)}), \\ \quad \vdots \\ \hat{s}_{n_R,n_N,k}^{(1)} = \mathcal{F}(r_{n_R,k}^{(1)}). \end{cases} \quad (6)$$

The structure of layer of the GMDH network is presented in Fig. 6.

As can be observed in the literature, most industrial systems are dynamic (Korbicz *et al.*, 2004; Patton *et al.*,

2000). Therefore, during system identification, it is natural to employ models which can represent the dynamics of the system. In the case of the GMDH neural network, the behaviour of each partial model should reflect that of the system being identified. It follows from the rule of the GMDH algorithm that the parameters of each partial model are estimated in such a way that its output is the best approximation of the real system output. In this situation, the partial model should have the ability to represent the dynamics. To tackle this problem, in this paper a dynamic neuron in the state-space representation is defined. The proposed dynamic neuron consists of two submodules: the linear state-space module and the activation module (Fig. 7).
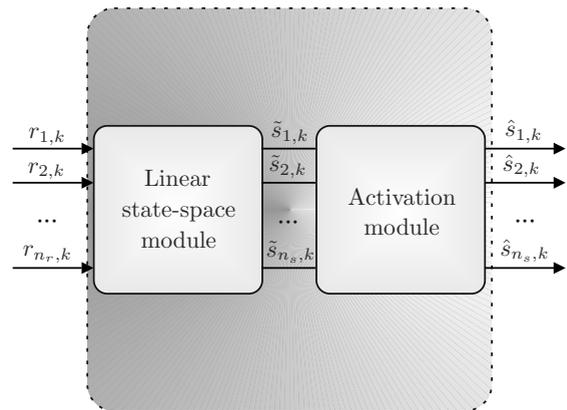
Fig. 7. Dynamic neuron model.

The behavior of the linear state-space part of a

dynamic neuron is described by the following equation:

$$\boldsymbol{z}_{k+1} = \boldsymbol{A}\boldsymbol{z}_k + \boldsymbol{B}\boldsymbol{r}_{i,k}, \tag{7}$$

$$\tilde{\boldsymbol{s}}_{i,j,k} = \boldsymbol{C}\boldsymbol{z}_k + \boldsymbol{D}\boldsymbol{r}_{i,k}, \tag{8}$$

where $\boldsymbol{r}_{i,k} \in \mathbb{R}^{n_r}$ and $\tilde{\boldsymbol{s}}_{i,j,k} \in \mathbb{R}^{n_s}$ are the inputs and outputs of the linear state-space submodule of the dynamic neuron. $\boldsymbol{A} \in \mathbb{R}^{n_z \times n_z}$, $\boldsymbol{B} \in \mathbb{R}^{n_z \times n_r}$, $\boldsymbol{C} \in \mathbb{R}^{n_s \times n_z}$, $\boldsymbol{D} \in \mathbb{R}^{n_s \times n_r}$, $\boldsymbol{z}_k \in \mathbb{R}^{n_z}$, where n_z represents the order of the dynamics (Ljung, 1999). Additionally, the matrix $\boldsymbol{A}$ has an upper-triangular form, i.e.,

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1,n_z} \\ 0 & a_{22} & \cdots & a_{2,n_z} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & a_{n_z,n_z} \end{bmatrix}. \tag{9}$$

This means that the dynamic neuron is asymptotically stable iff

$$|a_{i,i}| < 1, \quad i = 1, \ldots, n_z. \tag{10}$$

Moreover,

$$\boldsymbol{C} = \mathrm{diag}(c_1, \ldots, c_{n_s}, \underbrace{0, \ldots, 0}_{n_z - n_s}). \tag{11}$$

The linear state-space submodule output is used as the input for the activation module:

$$\hat{\boldsymbol{s}}_{i,j,k} = \mathcal{F}(\tilde{\boldsymbol{s}}_{i,j,k}), \tag{12}$$

with $\mathcal{F}(\cdot) = [f_1(\cdot), \ldots, f_{n_s}(\cdot)]^T$, where $f_i(\cdot)$ denotes a non-linear activation function (e.g., a hyperbolic tangent).

In order to estimate the unknown parameters of the dynamic neurons, the UKF (Teixeira *et al.*, 2010) can be applied. In the subsequent part of the paper, it will be shown that UKF-based constrained parameter estimation ensures asymptotically stable neurons of the GMDH model. Moreover, the application of this algorithm to the parameter estimation enables process to obtain the uncertainty of the partial models simultaneously. After the estimation, the parameters of the neurons are not modified during the further network synthesis. The obtained parameter estimates and their uncertainty enable calculation of the neuron responses and the adaptive threshold, which can be applied in the robust fault detection scheme. The details of such an algorithm will be presented in Section 4.

At the next stage of GMDH network synthesis, a validation data set $\mathcal{V}$ is used to calculate the processing error of each partial model in the current l-th network layer. The processing error $Q(\hat{\boldsymbol{s}}_{i,j}^{(l)})$ is usually calculated with the application of the evaluation criterion such as the *Final Prediction Error* (FPE), the *Akaike Information Criterion* (AIC) or the *F-test* (Ivakhnenko and Mueller,

1995). The evaluation of the processing errors Q for each neuron output is performed after the generation of the corresponding layer of neurons,

$$Q = \begin{bmatrix} Q(\hat{\boldsymbol{s}}_{1,1,k}^{(l)}) & \cdots & Q(\hat{\boldsymbol{s}}_{1,j,k}^{(l)}) & \cdots & Q(\hat{\boldsymbol{s}}_{1,n_N,k}^{(l)}) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ Q(\hat{\boldsymbol{s}}_{i,1,k}^{(l)}) & \cdots & Q(\hat{\boldsymbol{s}}_{i,j,k}^{(l)}) & \cdots & Q(\hat{\boldsymbol{s}}_{i,n_N,k}^{(l)}) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ Q(\hat{\boldsymbol{s}}_{n_R,1,k}^{(l)}) & \cdots & Q(\hat{\boldsymbol{s}}_{n_R,j,k}^{(l)}) & \cdots & Q(\hat{\boldsymbol{s}}_{n_R,n_N,k}^{(l)}) \end{bmatrix}, \tag{13}$$

Based on the defined evaluation criterion, it is possible to select the best-fitted neurons in the layer. Selection methods in GMDH neural networks play the role of a mechanism of structural optimization at the stage of constructing a new layer of neurons. According
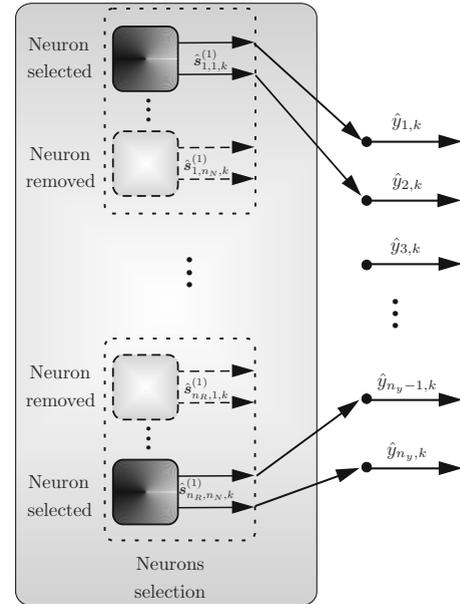
Fig. 8. Selection in the MIMO GMDH network.

to the chosen selection method, elements that introduce excessive processing error are removed (Fig. 8). One of the most interesting ways of performing the selection procedure is the application of the method based on the soft selection approach:

Input : Q—the matrix of the quality indexes of all dynamic neurons in the l-th layer, n_o—the number of opponent neurons, n_w—the number of winnings required for the i-th neuron selection.

Output : The set of neurons after selection.

(i) Select $j = 1$ column of matrix Q representing the quality indexes of all n_R neurons modeling the j-th vector of system outputs $\boldsymbol{s}_{i,j,k}$ created on the basis of all $i = 1, \ldots, n_R$ vectors of system inputs $\boldsymbol{r}_{i,k}$.

(ii) Conduct series of n_y competitions between each i-th neuron in the j-th column and n_o randomly

selected neurons (the so-called *opponent*) from the same column. The i-th neuron is the so-called *winner* neuron when

$$Q(\hat{\boldsymbol{s}}_{i,1,k}^{(l)}) \leq Q_o(\hat{\boldsymbol{s}}_{i,1,k}^{(l)}), \qquad (14)$$

where $o = 1, \ldots, n_o$ and Q_o denotes a quality index of the opponent neuron;

(iii) Select the neurons for the $(l + 1)$-th layer with the number of winnings bigger than n_w (the remaining neurons are removed);

(iv) Repeat the steps (1)–(3) for $j = 2, \ldots, n_N$ column of matrix Q representing the quality indexes of all neurons modeling the remaining $j = 2, \ldots, n_N$ vectors of system outputs $\hat{\boldsymbol{s}}_{i,1,k}^{(l)}$.

The main advantage of such an approach is the possibility to use potentially unfitted neurons which in the next layers may improve the quality of the model. One of the most important parameters which should be chosen in the selection process is the number of n_o opponents. A bigger value of n_o makes the probability of the selection of a neuron with a small quality index lower. In this way, in an extreme situation, when $n_o \gg n_y$, the soft selection method will only select the best fitted neurons.

After the selection procedure, the outputs of the selected neurons become the inputs to the neurons in the subsequent layer:

$$\begin{cases} u_{1,k}^{(l+1)} = \hat{y}_{1,k}^{(l)}, \\ u_{2,k}^{(l+1)} = \hat{y}_{2,k}^{(l)}, \\ \quad \vdots \\ u_{n_u,k}^{(l+1)}(k) = \hat{y}_{n_y,k}^{(l)}. \end{cases} \qquad (15)$$

During the synthesis of the GMDH neural network, the number of layers suitably increases. Each time a new layer is added, new neurons are successively introduced to the network (Fig. 9).

In the next stage of GMDH network synthesis the termination condition testing is performed. The synthesis of the GMDH network is completed when the network fits the data with a desired accuracy or the introduction of new neurons does not cause a significant increase in the approximation abilities of the neural network.

For this reason the quality indexes $Q(\hat{\boldsymbol{s}}_{i,j}^{(l)})$ for all neurons included in the l layer are calculated. The quality index $Q_{j,\min}^{(l)}$ represents the processing error for the best neuron in this layer which approximates the j-th vector of system outputs:

$$Q_{j,\min}^{(l)} = \min_{i=1,\ldots,n_R} Q(\hat{\boldsymbol{s}}_{i,j}^{(l)}) \quad \text{for} \quad j = 1, \ldots, n_N. \quad (16)$$

The values $Q(\hat{\boldsymbol{s}}_{i,j}^{(l)})$ can be determined with the application of the defined evaluation criterion, which was used in
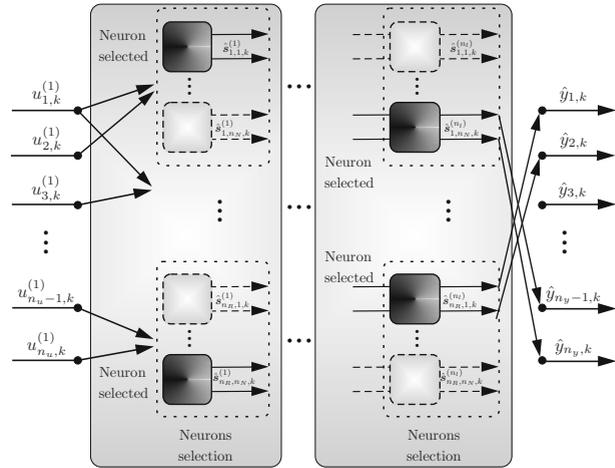
Fig. 9. Synthesis of the GMDH neural network.

the selection process. The synthesis of the network is completed when each of the calculated quality indexes reaches the minimum:

$$Q_{j,\min}^{(l_{opt})} = \min_{l=1,\ldots,n_l} Q_{j,\min}^{(l)} \quad \text{for} \quad j = 1, \ldots, n_N. \quad (17)$$
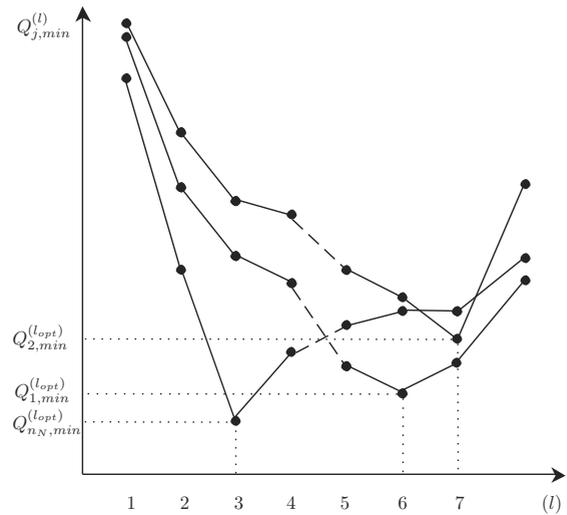
Fig. 10. Termination of the synthesis of the multi-output GMDH network network.

The termination of the synthesis occurs independently for each vector of system outputs $\hat{\boldsymbol{s}}_{i,j}^{(l)}$ and as a result a set of quality indexes, corresponding to each vector of system outputs, is obtained: $Q_1, Q_2, \ldots, Q_{n_N}$ (Fig. 10). A particular minimum can occur at different stages of network synthesis. This is why, in the multi-output network, outputs of the resulting structure are usually in different layers (Fig. 11).
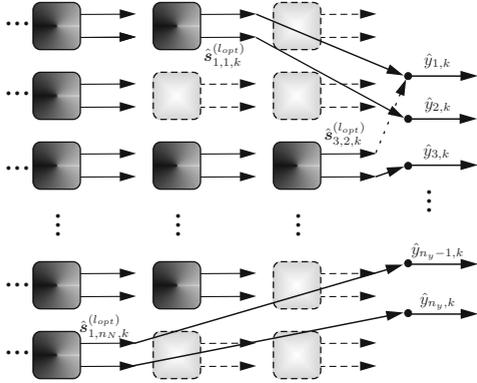
Fig. 11. Final structure of the MIMO GMDH neural network.

As a result of the process of GMDH network synthesis, the set of vectors of potential system outputs will be obtained. It may happen that in these vectors there will be a few outputs modeling the particular system output. In this case, the neuron output which characterizes the best modeling quality should be chosen.

4. Parameter estimation of dynamic neurons with the application of the unscented Kalman filter

In Section 3, it was mentioned that the parameters of each neuron in the GMDH neural network are estimated separately. This property allows applying the UKF in the process of synthesis of the GMDH neural network. It will be shown that the main advantage of application of the UKF to parameter estimation of the dynamic neurons with constraints ensures an asymptotically stable GMDH model. Moreover, the application of the UKF allows to calculate the model uncertainty, which then can be applied to the robust fault detection.

Let us define a state vector (in order to simplify the notation, the indexes $_i^{(l)}$ in the $r_{i,k}^{(l)}$ and $_{i,j}^{(l)}$ in $\hat{s}_{i,j,k}^{(l)}$ are omitted):

$$x_k = \begin{bmatrix} z_k \\ p_k \end{bmatrix}, \tag{18}$$

which is composed of the parameter vector of the neuron p_k as well as of the state of the neuron, which is described as

$$z_{k+1} = A(p_k)z_k + B(p_k)r_k, \tag{19}$$

$$\tilde{s}_k = C(p_k)z_k + D(p_k)r_k, \tag{20}$$

$$\hat{s}_k = \mathcal{F}(\tilde{s}_k). \tag{21}$$

The vector p_k is composed of the diagonal elements of the matrix A, i.e.,

$$p_k = [a_{11}, \ldots, a_{n,n}, \ldots]^T, \tag{22}$$

while the remaining elements of p_k are composed of the rest of the parameters of A, as well as all elements of B, C and D. Thus, the dimension of p_k is

$$\dim(p_k) = \frac{(n_z \times n_z) + n_z}{2} \\ + n_z \times n_r + n_s + n_s \times n_r. \tag{23}$$

It should also be pointed out that instead of A (B, C, D) the notation $A(p_k)$ $(B(p_k), C(p_k), D(p_k))$ is introduced which clearly denotes the dependence on p_k. Finally, the state-space model is

$$x_{k+1} = \begin{bmatrix} A(p_k)z_k + B(p_k)r_k \\ p_k \end{bmatrix} + \varpi_k = \\ \mathcal{G}(x_k, r_k) + \varpi_k, \tag{24}$$

$$s_k = \mathcal{F}(C(p_k)z_k + D(p_k)r_k) + v_k \\ = \mathcal{H}(x_k, r_k) + v_k, \tag{25}$$

where $\mathcal{G} : \mathbb{R}^n \times \mathbb{R}^{n_r} \to \mathbb{R}^n$ and $\mathcal{H} : \mathbb{R}^n \times \mathbb{R}^{n_r} \to \mathbb{R}^{n_s}$ are the process and observation models, respectively. $r_k \in \mathbb{R}^{n_r}$ and $s_k \in \mathbb{R}^{n_s}$ are the input and output data, $\rho(x_0)$, $\rho(\varpi_{k-1})$, $\rho(v_k)$ denote the *Probability Density Functions* (PDFs) of normal distributions, where $x_0 \in \mathbb{R}^n$ is the initial state vector, $\varpi_{k-1} \in \mathbb{R}^n$ is the process noise, and $v_0 \in \mathbb{R}^n$ is the measurement noise. It is assumed that the process noise and the measurement noise are uncorrelated. Moreover, the means and covariance matrices of $\rho(\varpi_{k-1})$ and $\rho(v_k)$ are known and equal to zero and Q, R, respectively.

The profit function which is the value of the conditional PDF of the state vector $x_k \in \mathbb{R}^n$ given the past and present measured data $s_1, \ldots, s_k$ is defined as follows:

$$J(x_k) \triangleq \rho(x_k|(s_1, \ldots, s_k)). \tag{26}$$

The parameter and state estimation problem can be defined as the maximization of (26). In order to solve the following problem, the UKF can be applied (Julier and Uhlmann, 2004). It employs the unscented transform (Julier and Uhlmann, 2004), which approximates the mean $\hat{s}_k \in \mathbb{R}^{n_s}$ and covariance $P_k^{ss} \in \mathbb{R}^{n_s \times n_s}$ of the random vector s_k obtained from the non-linear transformation $s_k = \mathcal{H}(x_k)$, where x_k is a random vector, whose mean $\hat{x}_k \in \mathbb{R}^n$ and covariance $P_k^{xx} \in \mathbb{R}^{n \times n}$ are assumed to be known.

The unscented transform is based on a set of deterministically chosen vectors $\mathcal{X}_{j,k} \in \mathbb{R}^n$, $j = 0, \ldots, 2n$, known as sigma points:

$$\hat{x}_k = \sum_{j=0}^{2n} w_j \mathcal{X}_{j,k}, \tag{27}$$

$$P_k^{xx} = \sum_{j=0}^{2n} w_j [\mathcal{X}_{j,k} - \hat{x}_k][\mathcal{X}_{j,k} - \hat{x}_k]^T, \tag{28}$$

with weights $\boldsymbol{w} \triangleq [w_0, w_1 \ldots, w_{2n}] \in \mathbb{R}^{2n+1}$ satisfying $\sum_{j=0}^{2n} w_j = 1$, where

$$w_0 \triangleq \frac{\lambda}{n+\lambda}, \quad w_j \triangleq \frac{1}{2(n+\lambda)}, \quad j = 1, \ldots, 2n, \quad (29)$$

and the sigma-point matrix $\mathcal{X}_k \triangleq [\mathcal{X}_{0,k}, \mathcal{X}_{1,k}, \ldots, \mathcal{X}_{2n,k}] \in \mathbb{R}^{n \times (2n+1)}$ is chosen as

$$\mathcal{X}_k = \hat{\boldsymbol{x}}_k \mathbf{1}_{1 \times (2n+1)} + \sqrt{n+\lambda} \\ \left[\mathbf{0}_{n \times 1} \ (\boldsymbol{P}_k^{xx})^{1/2} \ -(\boldsymbol{P}_k^{xx})^{1/2} \right], \quad (30)$$

where $(\cdot)^{1/2}$ is the Cholesky square root and $\lambda > -n$. Propagating each sigma point through $\mathcal{H}$ yields

$$\mathcal{S}_{j,k} = \mathcal{H}(\mathcal{X}_{j,k}, \boldsymbol{r}_k), \quad j = 0, \ldots, 2n, \quad (31)$$

where

$$\hat{\boldsymbol{s}}_k = \sum_{j=0}^{2n} w_j \mathcal{S}_{j,k}, \quad (32)$$

$$\boldsymbol{P}_k^{ss} = \sum_{j=0}^{2n} w_j [\mathcal{S}_{j,k} - \hat{\boldsymbol{s}}_k][\mathcal{S}_{j,k} - \hat{\boldsymbol{s}}_k]^T. \quad (33)$$

The unscented Kalman filter is a two-step estimator, where the first forecast step is performed according to the following equations:

$$\mathcal{X}_{j,k|k-1} = \mathcal{G}(\mathcal{X}_{j,k-1|k-1}, \boldsymbol{r}_{k-1,k-1}), \\ j = 0, \ldots, 2n, \quad (34)$$

where $\mathcal{G}$ is defined by the right-hand side of Eqn. (24).

$$\hat{\boldsymbol{x}}_{k,k-1} = \sum_{j=0}^{2n} w_j \mathcal{X}_{j,k|k-1}, \quad (35)$$

$$\boldsymbol{P}_{k,k-1}^{xx} = \sum_{j=0}^{2n} w_j [\mathcal{X}_{j,k|k-1} - \hat{\boldsymbol{x}}_{k,k-1}] \\ \times [\mathcal{X}_{j,k|k-1} - \hat{\boldsymbol{x}}_{k,k-1}]^T + \boldsymbol{Q}, \quad (36)$$

$$\mathcal{S}_{j,k|k-1} = \mathcal{H}(\mathcal{X}_{j,k|k-1}, k), \quad j = 0, \ldots, 2n, \quad (37)$$

$$\hat{\boldsymbol{s}}_{k|k-1} = \sum_{j=0}^{2n} w_j \mathcal{S}_{j,k|k-1}, \quad (38)$$

$$\boldsymbol{P}_{k|k-1}^{ss} = \sum_{j=0}^{2n} w_j [\mathcal{S}_{j,k|k-1} - \hat{\boldsymbol{s}}_{k|k-1}] \\ \times [\mathcal{S}_{j,k|k-1} - \hat{\boldsymbol{s}}_{k,k-1}]^T + \boldsymbol{R}, \quad (39)$$

$$\boldsymbol{P}_{k|k-1}^{xs} = \sum_{j=0}^{2n} w_j [\mathcal{X}_{j,k|k-1} - \hat{\boldsymbol{x}}_{k|k-1}] \\ \times [\mathcal{S}_{j,k|k-1} - \hat{\boldsymbol{s}}_{k,k-1}]^T, \quad (40)$$

where $\boldsymbol{P}_{k|k-1}^{xx}$ is the forecast error covariance, $\boldsymbol{P}_{k|k-1}^{ss}$ is the innovation covariance, $\boldsymbol{P}_{k|k-1}^{xs}$ is the cross covariance, and $\boldsymbol{P}_{k|k}^{ss}$ is the data-assimilation error-covariance.

The second step of the UKF is called the data-assimilation step and it is described by the following classical Kalman filter update equations:

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}^{xs} (\boldsymbol{P}_{k|k-1}^{ss})^{-1}, \quad (41)$$

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{s}_k - \hat{\boldsymbol{s}}_{k|k-1}), \quad (42)$$

$$\boldsymbol{P}_{k|k}^{xx} = \boldsymbol{P}_{k|k-1}^{xx} - \boldsymbol{K}_k \boldsymbol{P}_{k|k-1}^{ss} \boldsymbol{K}_k^T, \quad (43)$$

where $\boldsymbol{K}_k \in \mathbb{R}^{n \times n_s}$ is the Kalman gain matrix.

The task of training a dynamic neuron relies on the estimation of parameter vector $\boldsymbol{x}_k$ which satisfies the following interval constraint:

$$-1 + \delta \leq \boldsymbol{e}_i^T \boldsymbol{x}_k \leq 1 - \delta, \quad i = 1, \ldots, n, \quad (44)$$

where $\boldsymbol{e}_i \in \mathbb{R}^{n_p+n}$, whereas $\boldsymbol{e}_1 = [1, 0, \ldots, 0]^T$, $\boldsymbol{e}_2 = [0, 1, \ldots, 0]^T$, $\ldots$, $\boldsymbol{e}_{n_p+n} = [0, 0, \ldots, 1]^T$, and δ is a small positive value. These constraints follow directly from the asymptotic stability condition (10), while δ is introduced in order to make the above mentioned problem tractable.

The neural model has a cascade structure, which follows from the fact that the neuron outputs constitute the neuron inputs in the subsequent layers. The neural model which is a result of the cascade connection of dynamic neurons is asymptotically stable when each of the neurons is asymptotically stable (Lee and Jiang, 2006). Thus, the fulfilment of (10) (being a result of (44)) for each neuron allows obtaining an asymptotically stable dynamic GMDH neural model. Thus, the objective of the interval-constrained parameter-estimation problem is to maximize (26) subject to (44). In order to perform the neuron training process, it is necessary to truncate the probability density function at the n constraint edges given by the rows of the state interval constraint (44) such that the pseudomean $\hat{\boldsymbol{x}}_{k,k}^t$ of the truncated PDF is an interval-constrained state estimate with the truncated error covariance $\boldsymbol{P}_{k,k}^{xx}$. According to Teixeira *et al.* (2010), the normal distribution for the PDF truncation procedure is assumed.

The procedure of PDF truncation (Teixeira *et al.*, 2010) can be performed in i steps, where $i = 1, \ldots, n$. Let $\hat{\boldsymbol{x}}_{k|k,i}^t$ and $\boldsymbol{P}_{k|k,i}^{xxt}$ denote, respectively, the state estimate and the error covariance after enforcing the first $i = 1$ rows of the state interval constraint (44).

At the beginning, at $i = 1$ the initialization of $\hat{\boldsymbol{x}}_{k|k,i}^t = \hat{\boldsymbol{x}}_{k|k}$ and $\boldsymbol{P}_{k|k,i}^{xxt} = \boldsymbol{P}_{k|k}^{xx}$ according to (42) and (43) is performed. Next, for $i = 1, \ldots, n$, perform the following procedure:

Step 1. Find the orthogonal matrix $\boldsymbol{S} \in \mathbb{R}^{n \times n}$ and the diagonal matrix $\boldsymbol{D} \in \mathbb{R}^{n \times n}$ from the Schur

decomposition (Bernstein, 2005) of $\boldsymbol{P}_{k|k,i}^{xxt}$ given by $\boldsymbol{S}\boldsymbol{D}\boldsymbol{S}^T = \boldsymbol{P}_{k|k}^{xxt}$, where $\boldsymbol{P}_{k|k,i}^{xxt}$ is symmetric.

Step 2. Perform Gram–Schmidt orthogonalization to find the orthogonal matrix $\boldsymbol{\Theta} \in \mathbb{R}^{n \times n}$ satisfying

$$\boldsymbol{\Theta}\boldsymbol{D}^{1/2}\mathrm{col}_i(\boldsymbol{S}^T) = \left[\sqrt{\boldsymbol{P}_{(i,i),k|k,i}^{xxt}}\ \boldsymbol{0}_{1\times(n-1)}\right]^T, \quad (45)$$

which for $l = 1$ is given by

$$\mathrm{row}_l(\boldsymbol{\Theta}) = \frac{1}{\sqrt{\boldsymbol{P}_{(i,i),k|k,i}^{xxt}}}\mathrm{row}_i(\boldsymbol{S})\boldsymbol{D}^{1/2}, \quad (46)$$

and for $l = 2, \ldots, n$ by

$$\mathrm{row}_l(\boldsymbol{\Theta}) = (e_l - \sum_{q=1}^{l-1}(e_l^T\mathrm{col}_q(\boldsymbol{\Theta}^T))\mathrm{col}_q(\boldsymbol{\Theta}^T))^T, \quad (47)$$

where $e_l \triangleq \mathrm{col}_l(\boldsymbol{I}_{n\times n})$; if $\mathrm{row}_l(\boldsymbol{\Theta}) = \boldsymbol{0}_{1\times n}$, then reset

$$\mathrm{row}_l(\boldsymbol{\Theta}) = (e_1 - \sum_{q=1}^{l-1}(e_1^T\mathrm{col}_q(\boldsymbol{\Theta}^T))\mathrm{col}_q(\boldsymbol{\Theta}^T))^T. \quad (48)$$

Also, normalize for $l = 1, \ldots, n$

$$\mathrm{row}_l(\boldsymbol{\Theta}) = \frac{1}{\|\mathrm{row}_l(\boldsymbol{\Theta})\|_2}\mathrm{row}_l(\boldsymbol{\Theta}). \quad (49)$$

Step 3. Find the parameters of the interval constraint $a_{k,i} \leq \varsigma_{i,k,i} \leq b_{k,i}$, where $a_{k,i} < b_{k,i} \in \mathbb{R}$ are given by the following expressions:

$$a_{k,i} = \frac{1}{\sqrt{\boldsymbol{P}_{(i,i),k|k,i}^{xxt}}}(d_{i,k} - x_{i,k|k,i}^t), \quad (50)$$

$$b_{k,i} = \frac{1}{\sqrt{\boldsymbol{P}_{(i,i),k|k,i}^{xxt}}}(e_{i,k} - x_{i,k|k,i}^t), \quad (51)$$

and $\varsigma_{i,k} \triangleq \boldsymbol{\Theta}\boldsymbol{D}^{-1/2}\boldsymbol{S}^T(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k,i}^t) \in \mathbb{R}^n$ with mean

$$\hat{\varsigma}_{k,i} = [\mu_i\ \boldsymbol{0}_{1\times(n-1)}]^T, \quad (52)$$

and covariance

$$\boldsymbol{P}_{k,i}^{\varsigma\varsigma} = \mathrm{diag}(\sigma_i^2, \boldsymbol{1}_{1\times(n-1)}), \quad (53)$$

where

$$\alpha_i = \frac{\sqrt{2}}{\sqrt{\pi}[\mathrm{erf}(\frac{b_{k,i}}{\sqrt{2}}) - \mathrm{erf}(\frac{a_{k,i}}{\sqrt{2}})]}, \quad (54)$$

$\mathrm{erf}(\cdot)$ being the error function defined as

$$\mathrm{erf}(t) \triangleq \frac{2}{\sqrt{\pi}}\int_0^t \exp(-\tau^2)\,\mathrm{d}\tau, \quad (55)$$

and

$$\mu_i = \alpha_i\left[\exp\left(\frac{-a_{k,i}^2}{2}\right) - \exp\left(\frac{-b_{k,i}^2}{2}\right)\right], \quad (56)$$

$$\begin{aligned}\sigma_i^2 =\alpha_i\Big[&\exp\left(\frac{-a_{k,i}^2}{2}\right)(a_{k,i} - 2\mu_i)\\ &- \exp\left(\frac{-b_{k,i}^2}{2}\right)(b_{k,i} - 2\mu_i))\Big] + \mu_i^2 + 1.\end{aligned} \quad (57)$$

Step 4. Perform the inverse transformation

$$\hat{\boldsymbol{x}}_{k|k,i+1}^t = \boldsymbol{S}\boldsymbol{D}^{1/2}\boldsymbol{\Theta}^T\hat{\varsigma}_{k,i} + \hat{\boldsymbol{x}}_{k|k,i}^t, \quad (58)$$

$$\boldsymbol{P}_{k|k,i+1}^{xx,t} = \boldsymbol{S}\boldsymbol{D}^{1/2}\boldsymbol{\Theta}^T\boldsymbol{P}_{k,i}^{\varsigma\varsigma}\boldsymbol{\Theta}\boldsymbol{D}^{1/2}\boldsymbol{S}^T, \quad (59)$$

and write i instead of n, obtaining $\hat{\boldsymbol{x}}_{k|k}^t = \hat{\boldsymbol{x}}_{k|k,n+1}^t$ and $\boldsymbol{P}_{k|k}^{xxt} = \boldsymbol{P}_{k|k,n+1}^{xxt}$.

The probability density function truncation procedure allows avoiding the explicit on-line solution of a constrained optimization problem at each time step. Moreover, it assimilates the interval-constraint information in the state estimate $\hat{\boldsymbol{x}}_{k|k}^t$ and the error covariance $\boldsymbol{P}_{k|k}^{xxt}$.

The application of the UKF allows obtaining the state estimates as well as the uncertainty of the GMDH model in the form of a matrix $\boldsymbol{P}^{sst}$, which can then be applied to the calculation of the system output adaptive threshold (Fig. 12) and permits to perform robust fault detection:

$$\hat{y}_{i,k}^m = \mathcal{F}_i\left(\boldsymbol{c}_i\hat{\boldsymbol{x}}_k - t_{n_t-n_p}^{\alpha/2}\hat{\sigma}_i\sqrt{\boldsymbol{c}_i\boldsymbol{P}^{xxt}\boldsymbol{c}_i^T}\right), \quad (60)$$

$$\hat{y}_{i,k}^M = \mathcal{F}_i\left(\boldsymbol{c}_i\hat{\boldsymbol{x}}_k + t_{n_t-n_p}^{\alpha/2}\hat{\sigma}_i\sqrt{\boldsymbol{c}_i\boldsymbol{P}^{xxt}\boldsymbol{c}_i^T}\right), \quad (61)$$

where $\boldsymbol{c}_i$ stands for the i-th row ($i = 1, \ldots, n_s$) of the matrix $\boldsymbol{C}$ of the output neuron, $\hat{\sigma}_i$ is the standard deviation of the i-th fault-free residual and $t_{n_t-n_p}^{\alpha/2}$ is the t-Student distribution quantile.
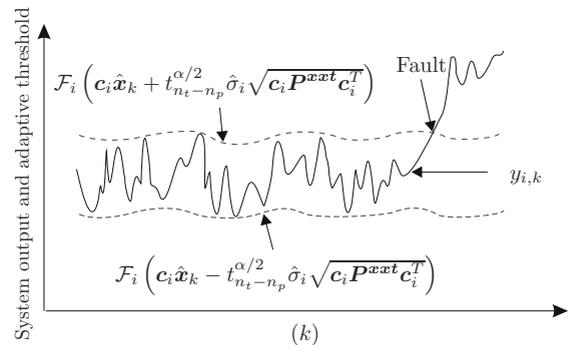
Fig. 12. Application of the adaptive threshold to robust fault detection.

5. Experimental results

The objective of this section is to design a dynamic GMDH model according to the approaches described in the previous sections and apply it robust fault detection of a tunnel furnace (Fig. 13). The tunnel furnace considered

Fig. 13. Tunnel furnace.

is designed to mimic, in laboratory conditions, real industrial tunnel furnaces, which can be applied in the food industry or production of ceramics, among others. The furnace is equipped with three electric heaters and four temperature sensors. The required temperature of the furnace can be kept by controlling the heaters' behaviour. This task can be achieved by group regulation of the voltage with the application of the PACSystems RX3i controller manufactured by GE Fanuc Intelligent Platforms and RP6 semiconductor relays produced by LUMEL. The temperature of the furnace is measured via the IC695ALG600 module with Pt100 Resistive Thermal Devices (RTDs). The visualisation of the behaviour of the tunnel furnace is made by the Quickpanel CE device from GE Fanuc Intelligent Platforms. The tunnel furnace can be considered a three-input and four-output system $(t_1, t_2, t_3, t_4) = f(u_1, u_2, u_3)$, where $t_1, \ldots, t_4$ represent measurements of the temperatures from four subsequent sensors and $u_1, \ldots, u_3$ denote the input voltages allowing to control the heaters.

The input and output data used for the identification and validation were collected in two data sets consisting of 2600 samples. It should also be pointed out that these data sets were scaled for the purpose of neural network design. 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 a kind of transformation, linear scaling can be used. Moreover, the data sets used for identification and validation were filtered with the application of second order Butterworth filters. Furthermore, the offset levels from data sets were removed.

The parameters of dynamic neurons (proposed in Section 3) were estimated with the application of the training algorithm, which is based on the UKF approach presented in Section 4. The selection of the best performing neurons in terms of their processing accuracy was realized with the application of the soft selection

method based on the sum squared error evaluation criterion. Table 1 presents the values of the evolution criterion for the subsequent layers, i.e., these values are obtained for the best performing neurons in a particular layer of the GMDH neural network. The results show that

Table 1. Values of the evaluation criterion $Q_{\mathcal{V}}(\hat{t}_1) - Q_{\mathcal{V}}(\hat{t}_4)$ for the best neurons in the subsequent layers for the validation data set.

Layer	$Q_{\mathcal{V}}(\hat{t}_1)$	$Q_{\mathcal{V}}(\hat{t}_2)$	$Q_{\mathcal{V}}(\hat{t}_3)$	$Q_{\mathcal{V}}(\hat{t}_4)$
1	0.0174	0.0154	0.0401	0.0212
2	0.0321	0.0136	0.0329	0.0188
3	0.0007	0.0006	0.0091	0.0012
4	0.0005	0.0003	0.0007	0.0009
5	0.0071	0.0010	0.0141	0.0054

gradual decrease of the value of the evaluation criterion occurs when a new layer of the GMDH network is introduced. It follows from the increase of the model complexity as well as its modelling abilities. However, when the model is too complex, the quality index $Q_{\mathcal{V}}$ increases. This situation occurs when the 5-th layer of the network is added. This means that the model should have only four layers.

Additionally, for comparison, the results obtained with the application of the linear state-space model are presented (cf. Table 2). In particular, as a result of using models within the range from 1 up to 10, the 5-th order state-space model was applied. Figure 14 shows

Table 2. Values of the evaluation criterion $Q_{\mathcal{V}}(\hat{t}_1) - Q_{\mathcal{V}}(\hat{t}_4)$ for the non-linear dynamic GMDH and linear state-space models for the validation data set.

Model	$Q_{\mathcal{V}}(\hat{t}_1)$	$Q_{\mathcal{V}}(\hat{t}_2)$	$Q_{\mathcal{V}}(\hat{t}_3)$	$Q_{\mathcal{V}}(\hat{t}_4)$
Linear	0.0060	0.0037	0.0040	0.0027
GMDH	0.0005	0.0003	0.0007	0.0009

residual signals calculated as the difference between temperature t_1 of the tunnel furnace and the linear state-space model and the non-linear dynamic GMDH model, respectively. The results show that the quality of the non-linear dynamic GMDH model is better than that of the linear state-space model. It follows from the non-linear nature of the identified tunnel furnace. Figures 15–18 show the temperatures $t_1 - t_4$ of the furnace and the adaptive thresholds obtained with Eqns. (60)–(61) for the validation data set (no fault case).

After the synthesis of the GMDH model, it is possible to employ it for robust fault detection of the tunnel furnace. Figure 19 presents the measurements of the temperature t_1 from the faulty sensor (simulated during 10 seconds) and the adaptive threshold obtained
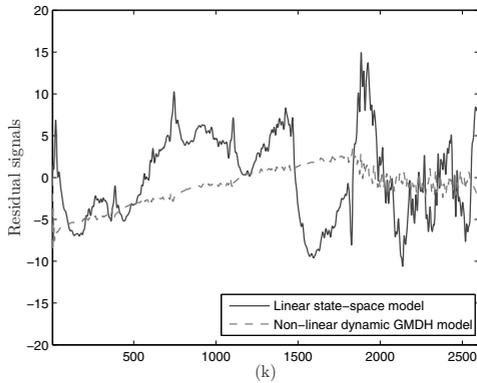
Fig. 14. Residual signals for the temperature t_1 of the tunnel furnace and the linear state-space model and the non-linear dynamic GMDH model.

Fig. 15. Temperature t_1 of the furnace and the corresponding adaptive threshold obtained with the GMDH model.

Fig. 16. Temperature t_2 of the furnace and the corresponding adaptive threshold obtained with the GMDH model.

Fig. 17. Temperature t_3 of the furnace and the corresponding adaptive threshold obtained with the GMDH model.

Fig. 18. Temperature t_4 of the furnace and the corresponding adaptive threshold obtained with the GMDH model.
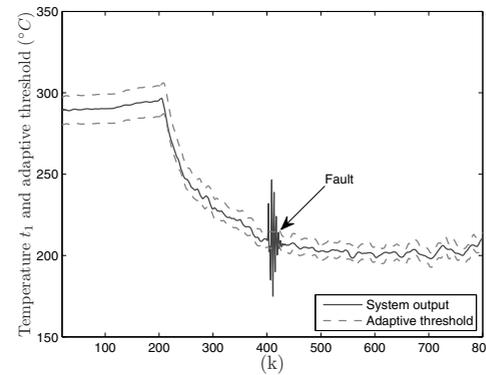
Fig. 19. Detection of the faulty temperature sensor with the application of the adaptive threshold.

with the application of the GMDH neural network. As can be seen a fault is detected for $k = 400$ when the value of the temperature t_1 crosses the adaptive threshold.

Figure 20 presents the detection of fault (starting from $k = 400$ to $k = 800$ seconds) simulated in the first electric heater of the tunnel furnace. As can be seen, the fault is detected when the measurements of the temperature t_1 drop rapidly and cross the adaptive

threshold obtained with the application of the GMDH neural model.

6. Conclusions

The objective of this paper was the design of a robust fault detection system based on the dynamic neural model. To tackle this problem, a GMDH neural network consisting
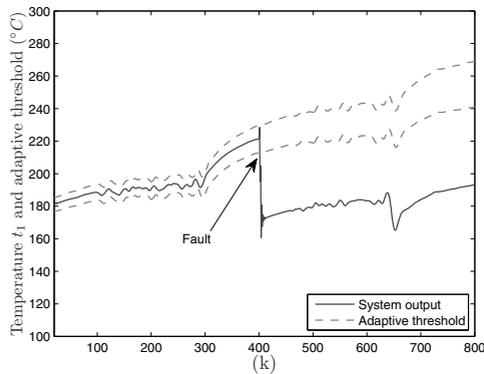
Fig. 20. Detection of the faulty electric heater with the application of the adaptive threshold.

of dynamic multi-input multi-output neurons in the state-space representation was developed. The state-space representation allows defining the stability conditions for each dynamic neuron and the whole GMDH network. Taking into account these conditions during parameter estimation with the application of the unscented Kalman filter algorithm a stable dynamic non-linear GMDH neural model can be obtained. Moreover, the application of the unscented Kalman filter allows obtaining the uncertainty of the GMDH neural model. This knowledge enables us to calculate the adaptive threshold of the GMDH model and apply it to robust fault detection of dynamic systems.

In the experimental part of the paper the results of application of the proposed approach to identification and robust fault detection of a tunnel furnace were presented. From the obtained results it can be noticed that the proposed dynamic non-linear GMDH neural model can be effectively applied to the identification of the tunnel furnace. Moreover, the comparison of the results obtained with the application of the proposed approach and the linear state-space model shows that the GMDH neural model can be successfully applied in the identification of dynamic non-linear system tasks. Finally, the resulting neural model was successfully used to design a robust fault detection scheme.

Acknowledgment

References

Back, A. and Tsoi, A. (1991). FIR and IIR synapses. A new neural network architectures for time series modelling, *Neural Computation* **3**(3): 375–385.

Bernstein, D. (2005). *Matrix Mathematics*, Princeton University Press, Princeton, NJ.

Ding, S. (2008). *Model-based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*, Springer-Verlag, Berlin/Heidelberg.

Elman, J. (1990). Finding structure in time, *Cognitive Science* **14**(2): 179–211.

Fasconi, P., Gori, M. and Soda, G. (1992). Local feedback multilayered networks, *Neural Computation* **4**(1): 120–130.

Gori, M., Bengio, Y. and De Mori, R. (1989). BPS: A learning algorithm for capturing the dynamic nature of speech, *International Joint Conference on Neural Networks, Washington, DC, USA*, pp. 417–423.

Gupta, M., Liang, J. and Homma, N. (2003). *Static and Dynamic Neural Networks*, John Wiley & Sons, Hoboken, NJ.

Haykin, S. (2001). *Kalman Filtering and Neural Networks*, John Wiley & Sons, New York, NY.

Haykin, S. (2009). *Neural Networks and Learning Machines*, Prentice Hall, New York, NY.

Ichalal, D., Marx, B., Ragot, J. and Maquin, D. (2012). New fault tolerant control strategies for nonlinear Takagi–Sugeno systems, *International Journal of Applied Mathematics and Computer Science* **22**(1): 197–210, DOI: 10.2478/v10006-012-0015-8.

Isermann, R. (2005). *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*, Springer-Verlag, Heidelberg/Berlin.

Ivakhnenko, A. and Mueller, J. (1995). Self-organization of nets of active neurons, *System Analysis Modelling Simulation* **20**(1–2): 93–106.

Jordan, M. and Bishop, C. (1997). Neural networks, *in* A. Tucker (Ed.), *CRC Handbook of Computer Science*, CRC Press, Boca Raton, FL, pp. 536–556.

Julier, S. and Uhlmann, J. (2004). Unscented filtering and nonlinear estimation, *Proceedings of the IEEE* **92**(3): 401–422.

Korbicz, J. and Kościelny, J. (Eds.) (2010). *Modeling, Diagnostics and Process Control: Implementation in the DiaSter System*, Springer-Verlag, Berlin.

Korbicz, J., Kościelny, J., Kowalczuk, Z. and Cholewa, W. (Eds.) (2004). *Fault Diagnosis. Models, Artificial Intelligence, Applications*, Springer-Verlag, Berlin.

Korbicz, J. and Mrugalski, M. (2008). Confidence estimation of GMDH neural networks and its application in fault detection system, *International Journal of System Science* **39**(8): 783–800.

Lee, T. and Jiang, Z. (2006). On uniform global asymptotic stability of nonlinear discrete-time systems with applications, *IEEE Transactions on Automatic Control* **51**(10): 1644–1660.

Ljung, L. (1999). *System Identification: Theory for the User*, Prentice Hall PTR, Upper Saddle River, NJ.

Montes de Oca, S., Puig, V., Witczak, M. and Dziekan, Ł. (2012). Fault-tolerant control strategy for actuator faults using LPV techniques: Application to a two degree of

freedom helicopter, *International Journal of Applied Mathematics and Computer Science* **22**(1): 161–171, DOI: 10.2478/v10006-012-0012-y.

Mrugalski, M. and Korbicz, J. (2007). Least mean square vs. outer bounding ellipsoid algorithm in confidence estimation of the GMDH neural networks, *in* B. Beliczynski, A. Dzielinski, M. Iwanowski, and B. Ribeiro (Eds.), *Adaptive and Natural Computing Algorithms, Part 2,* Lecture Notes in Computer Science, Vol. 4432, Physica-Verlag, Heidelberg, pp. 19–26.

Mrugalski, M. and Korbicz, J. (2011). GMDH neural networks, *in* B. Wilamowski and J. Irwin (Eds.), *The Industrial Electronics Handbook*, 2nd Edn., Vol. 5, CRC Press, Taylor Francis Group, Boca Raton, FL, pp. 8-1–8-21.

Mrugalski, M. and Witczak, M. (2012). State-space GMDH neural networks for actuator robust fault diagnosis, *Advances in Electrical and Computer Engineering* **12**(3): 65–72.

Mrugalski, M., Arinton, E. and Korbicz, J. (2003). Dynamic GMDH type neural networks, *in* L. Rutkowski and J. Kacprzyk (Eds.), *Neural Networks and Soft Computing*, Heidelberg, Physica-Verlag, pp. 698–703.

Mrugalski, M., Witczak, M. and Korbicz, J. (2008). Confidence estimation of the multi-layer perceptron and its application in fault detection systems, *Engineering Applications of Artificial Intelligence* **21**(6): 895–906.

Niemann, H.H. (2012). A model-based approach to fault-tolerant control, *International Journal of Applied Mathematics and Computer Science* **22**(1): 67–86, DOI: 10.2478/v10006-012-0005-x.

Noura, H., Theilliol, D., Ponsart, J. and Chamseddine, A. (2009). *Fault-tolerant Control Systems: Design and Practical Applications*, Springer-Verlag, London.

Palade, V., Bocaniala, C. and Jain, L. (2006). *Computational Intelligence in Fault Diagnosis*, Springer-Verlag, London.

Pan, Y., Sung, S. and Lee, J. (2001). Data-based construction of feedback-corrected nonlinear prediction model using feedback neural networks, *Control Engineering Practice* **9**(8): 859–867.

Patan, K., Witczak, M. and Korbicz, J. (2008). Towards robustness in neural network based fault diagnosis, *International Journal of Applied Mathematics and Computer Science* **18**(4): 443–454, DOI: 10.2478/v10006-008-0039-2.

Patton, R., Frank, P. and Clark, R. (2000). *Non-linear Systems Identification. From Classical Approaches to Neural Networks and Fuzzy Models*, Springer-Verlag, Berlin.

Teixeira, B., Torres, L., Aguirre, L. and Bernstein, D. (2010). On unscented Kalman filtering with state interval constraints, *Journal of Process Control* **20**(1): 45–57.

Williams, R. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks, *Neural Computation* **1**(2): 270–280.

Witczak, M. (2007). *Modelling and Estimation Strategies for Fault Diagnosis of Non-Linear Systems. From Analytical to Soft Computing Approaches*, Springer-Verlag, Berlin.

Witczak, M., Korbicz, J., Mrugalski, M. and Patton, R. (2006). A GMDH neural network based approach to robust fault detection and its application to solve the Damadics benchmark problem, *Control Engineering Practice* **14**(6): 671–683.

Witczak, M. and Prętki, P. (2007). Design of an extended unknown input observer with stochastic robustness techniques and evolutionary algorithms, *International Journal of Control* **80**(5): 749–762.

Zamarreño, J. M. and Vega, P. (1998). State space neural network. Properties and application, *Neural Networks* **11**(6): 1099–1112.

Marcin Mrugalski was born in Poland 1975. He received the M.Sc. and Ph.D. degrees in electrical engineering from the University of Zielona Góra, Poland, in 1999 and 2004, respectively. In 2004, he was with the University of Hull, UK, as a research fellow. He has been an assistant professor of automatic control and robotics at the Institute of Control and Computation Engineering, University of Zielona Góra, since 2004. His current research interests include neural networks, computational intelligence and fault diagnosis. He has taken part in the realization of a number of research projects sponsored by the Ministry of Scientific Research and Information Technology in Poland and by the European Commission: INCO-Copernicus (1997–1999) and 5th FP EU RTN (2000–2004). He has published more than 45 papers in international journals, book chapters and conference proceedings.