BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES
Vol. 54, No. 1, 2006

# Robust fault detection using analytical and soft computing methods

J. KORBICZ*

Institute of Control and Computation Engineering, University of Zielona Góra, 50 Podgórna St., 65-246 Zielona Góra, Poland

**Abstract.** The paper focuses on the problem of robust fault detection using analytical methods and soft computing. Taking into account the model-based approach to Fault Detection and Isolation (FDI), possible applications of analytical models, and first of all observers with unknown inputs, are considered. The main objective is to show how to employ the bounded-error approach to determine the uncertainty of soft computing models (neural networks and neuro-fuzzy networks). It is shown that based on soft computing models uncertainty defined as a confidence range for the model output, adaptive thresholds can be described. The paper contains a numerical example that illustrates the effectiveness of the proposed approach for increasing the reliability of fault detection. A comprehensive simulation study regarding the DAMADICS benchmark problem is performed in the final part.

**Key words:** fault detection, robustness, unknown input observer, neural networks, neuro-fuzzy systems, bounded-error approach, model uncertainty.

## 1. Introduction

All systems known in nature (physical, biological, and engineering ones) can malfunction and fail due to faults in their components. The possibilities of failures increase with systems' complexity, which is typical for modern engineering systems. The growing complexity of industrial installations, i.e., in chemical, petrochemical and power industries creates serious problems for process control by operators and, moreover, the degree of automation increases as well. From this point of view, more attention has to be paid to realiability, safety and fault tolerance in the design and operation of industrial installations.

In automatic control systems, defects may occur in sensors, actuators, components of the controlled process, or within the hardware or/and software of the control framework. Moreover, faults in a component may develop into failures of the whole system and this effect can easily be amplified by the closed loop. Therefore, fault tolerance of automatic control systems has gained more and more importance in the last decade [1].

The tolerance to faults can be achieved by different strategies, but the most important and difficult problem is early diagnosis of faults. Therefore, fault diagnosis has become an important issue in modern control theory and practice. During the last two and half decades, a huge amount of research has been conducted in this field and a great variety of methods increasingly accepted in practice have been proposed.

The most efficient fault-diagnostic strategy is the so-called model-based approach [2–6], in which either analytical or artificial intelligence models or a combination of both along with analytical or heuristic reasoning is applied. The main difficulty with applying analytical models is the fact that imprecise mathematical models are generally available. Therefore, the most essential requirement for analytical model-based fault detection and isolation is to provide robustness to different kinds of unmodelled disturbances and modelling errors. At present, different efficient robust FDI techniques [7–9] are proposed. Undoubtedly, the most common one is to use robust observers, such as the Unknown Input Oserver (UIO) [7,10,11], which can tolerate a degree of model uncertainty and hence increase the reliability of fault diagnosis.

There are, of course, many different observers which can be applied to non-linear, and especially non-linear deterministic systems. Logically, the number of "real world" applications (not only simulated examples) should proliferate, yet this is not the case. It seems that there are two main reasons why strong formal methods are not accepted in engineering practice. First, the design complexity of most observers for non-linear systems [7, 12] does not encourage engineers to apply them in practice. Second, the application of observers is limited by the need for non-linear state-space models of the system being considered, which is usually a serious problem in complex industrial systems.

To overcome these problems, the so-called soft computing methods (neural networks, neuro-fuzzy models, etc.) are being investigated, which can model a much wider classes of non-linear systems. Mathematical models used in the traditional FDI methods are potentially sensitive to modelling errors, parameter variation, noise and disturbances [8,13]. Process modelling has limiations, especially when the system is uncertain and the data are ambiguous. Soft computing (SC) methods (e.g., neural networks, fuzzy logic, and evolutionary algorithms) are known to overcome these problems to some extent [3,14,15]. Neural networks [16] are known for their approximation, generalization and adaptive capabilities and they can be very useful when analytical models are not available. However, the neural network operates as a black box with no qualitative information available. Fuzzy logic systems, on the

*e-mail: j.korbicz@issi.uz.zgora.pl

other hand, have the ability to model a non-linear system and to express it in the form of linguistic rules making it more transparent (i.e., easier to interprent). They also have inherent abilities to deal with imprecise or noisy data. A Neuro-Fuzzy (NF) model [17,18] can be used as a powerful combination of neural-networks and fuzzy logic techniques.

In this paper, we briefly review the fundamentals of model-based FDI and then focus our attention on some analytical and soft computing models used in diagnostic systems. As for analaytical methods, the main focus is on observers, including observers with unknown inputs. Then the adaptive threshold technique is employed to implement robust neural or nero-fuzzy model-based fault detection systems. This technique is based on the uncertainty of models defined as a confidence range for model outputs. The main attention will be paid to the modification and adaptation of the Bounded-Error Approach (BEA) for determinig the uncertainty of Group Method of Data Handling (GMDH) neural networks and Takagi-Sugeno NF systems. The main advantage of this approach is that it does not consider strong assumptions about the kind of noise like, for example, statistical methods [19]. It assumes only that bounds on the noise are available [20,21]. Experimental results presented in the final part of the paper confirm the effectiveness of the proposed approach for robust FDI using neural or neuro-fuzzy models. The DAMADICS benchmark problem is considered in this part.

## 2. Structure of a diagnosis system

The basic idea of the model-based approach to FDI is to compare the behaviour of the actual system with that of a functional system model. The traditional approach is to use analytical models and to check the model outputs for consistency with the measured outputs of the actual system. In general, the FDI task is accomplished by a two-step procedure (Fig. 1), which consists of residual generation and evaluation.

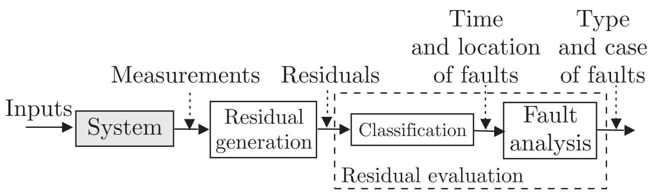


Fig. 1. Two-step procedure of the diagnosis process

In other words, three phases are distinguished in the process diagnosis [2,4,5,22]: detection, isolation and identification. The task of detection is to infer the occurrence of faults from residuals, and that of isolation – to define their location and time. Then the task of identification is to define the type, size and case of faults.

The general model-based diagnosis system is shown in Fig. 2. Note that the core of model-based FDI is a process model which has to be accurate. Otherwise, false alarms occur that falsify the results and make the FDI system useless. Residual evaluation is a logical decision-making process which transforms quantitative knowledge (residuals) into qualitative statements of the yes or no type.

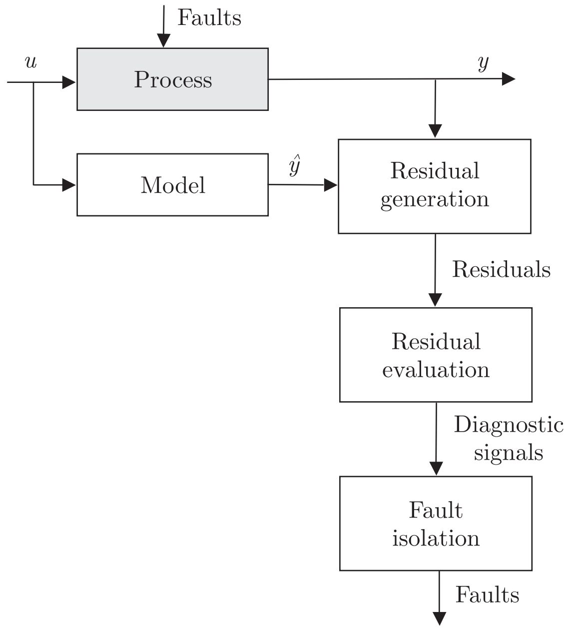


Fig. 2. General scheme of the model-based FDI system

## 3. Analytical models in fault detection systems

Analytical models most often used in FDI systems are given in the input-output (continuous or discrete transfer matrices) or the state space format. The dynamic linear continuous system can be given by the following state space equations [2,5]:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{R}_1\boldsymbol{f}(t), \tag{1}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) + \boldsymbol{R}_2\boldsymbol{f}(t), \tag{2}$$

where $t$ is the continuous time, $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the system state vector, $\boldsymbol{A}$ is the system matrix, $\boldsymbol{u}(t) \in \mathbb{R}^p$ is the known input vector, $\boldsymbol{B}$ is the input distribution matrix, $\boldsymbol{f}(t) \in \mathbb{R}^s$ is the vector of faults which can occur in actuators, sensors and components of the plant. The matrices $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are known and denote the corresponding fault distribution matrices, $\boldsymbol{y}(t) \in \mathbb{R}^m$ is the measurement vector, $\boldsymbol{C}$ and $\boldsymbol{D}$ are the known output matrices. The state space description (1)–(2) can be given for discrete time systems as well as by applying difference equations.

The corresponding multi-dimensional input-output model in the frequency domain is given by [23]:

$$\boldsymbol{y}(s) = \boldsymbol{G}_u(s)\boldsymbol{u}(s) + \boldsymbol{G}_f(s)\boldsymbol{f}(s), \tag{3}$$

where $s$ is the differential or shift operator, depending on whether the system is continuous or discrete, $\boldsymbol{G}_u(s)$ is the transfer matrix operator from $\boldsymbol{u}(s)$ to $\boldsymbol{y}(s)$, and $\boldsymbol{G}_f(s)$ is the fault transfer matrix operator from $\boldsymbol{f}(s)$ to $\boldsymbol{y}(s)$.

In Equations (1)–(3) it was assumed that the state space and input-output models are accurate, which is a very strong assumption from the practical point of view. Therefore, it is important to take into account in such models different modelling uncertainties such as parameter deviations, unmodelled dynamics and non-linearities as well as omitted inputs and different instances noise [8,23]. All of the mentioned uncertainties are the so-called unknown inputs $\boldsymbol{d}(t) \in \mathbb{R}^r$ and can be included in the model Eqs. (1)–(2) or (3). From the modelling point of view, full identification of unknown inputs $\boldsymbol{d}(t)$ is not needed, but they do have to be distinguished from faults. This is needed to avoid false alarms.

Taking into account modelling uncertainties given by unknown inputs and parameters deviations, the complete state space model used for residual generation can be described by

$$\dot{\boldsymbol{x}}(t) = (\boldsymbol{A}+\Delta\boldsymbol{A})\boldsymbol{x}(t)+(\boldsymbol{B}+\Delta\boldsymbol{B})\boldsymbol{u}(t)+\boldsymbol{Q}_1\boldsymbol{d}(t)+\boldsymbol{R}_1\boldsymbol{f}(t), \tag{4}$$

$$\boldsymbol{y}(t) = (\boldsymbol{C}+\Delta\boldsymbol{C})\boldsymbol{x}(t)+(\boldsymbol{D}+\Delta\boldsymbol{D})\boldsymbol{u}(t)+\boldsymbol{Q}_2\boldsymbol{d}(t)+\boldsymbol{R}_2\boldsymbol{f}(t), \tag{5}$$

where $\boldsymbol{d}(t)$ is an unknown input vector with known distribution matrices $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$, $\Delta\boldsymbol{A}$, $\Delta\boldsymbol{B}$, $\Delta\boldsymbol{C}$ and $\Delta\boldsymbol{D}$ represent modelling errors caused by parameter errors or deviations and appear as multiplicative unknown inputs.

The corresponding input-output model with modelling uncertaities is given by the Eq.

$$\boldsymbol{y}(s) = \big(\boldsymbol{G}(s) + \Delta\boldsymbol{G}_u(s)\big)\boldsymbol{u}(s) + \boldsymbol{G}_d(s)\boldsymbol{d}(s) + \boldsymbol{G}_f(s)\boldsymbol{f}(s), \tag{6}$$

where $\boldsymbol{G}_d(s)$ denotes the transfer matrix operator from $\boldsymbol{d}(s)$ to $\boldsymbol{y}(s)$, and $\Delta\boldsymbol{G}_u(s)$ is the transfer matrix operator which comprises both parametric faults and parameter errors of the model.

**3.1. Residual generators.** The core module of the model-based FDI system is the generation of residuals. In general, the structure of a residual generator can be given by [23]:

$$\boldsymbol{r}(s) = \boldsymbol{H}_u(s)\boldsymbol{u}(s) + \boldsymbol{H}_y(s)\boldsymbol{y}(s), \tag{7}$$

where $\boldsymbol{H}_u(s)$ and $\boldsymbol{H}_y(s)$ are realizable transfer matrices. According to this assumption, the residual $\boldsymbol{r}(t)$ should be equal to zero for the fault-free case, and different from zero otherwise, i.e.,

$$\boldsymbol{r}(t) \neq 0 \quad \text{iff} \quad f_i(t) \neq 0, \quad i = 1, 2, \ldots, q. \tag{8}$$

To satisfy the condition (8), two transfer matrices in (7) should be governed by [7]:

$$\boldsymbol{H}_u(s) + \boldsymbol{H}_y(s)\boldsymbol{G}_u(s) = 0. \tag{9}$$

Different forms of the residual generator can be obtained in a simple way by a proper choice of the transfer matrices $\boldsymbol{H}_u(s)$ and $\boldsymbol{H}_y(s)$. Taking into account the condition (9) and the model (6), the general form of the residual generator can be given by the relation

$$\boldsymbol{r}(s) = \boldsymbol{H}_y(s)\big[\Delta\boldsymbol{G}_u(s)\boldsymbol{u}(s) + \boldsymbol{G}_d(s)\boldsymbol{d}(s) + \boldsymbol{G}_f(s)\boldsymbol{f}(s)\big], \tag{10}$$

which includes all kinds of model uncertainties, i.e., $\Delta\boldsymbol{G}_u$ and $\Delta\boldsymbol{G}_d$. From (10) it follows that both faults and modelling uncertainty affect the residual, and hence there is a need to develop robust FDI algorithms. The essence of robust FDI is to discriminate between faults and uncertainties, i.e., a robust system should be sensitive to faults and insensitive to uncertainty [7,8]. Below the main classical methods that can be applied for the purpose of fault diagnosis are considered. The attention is restricted to two groups of approaches, i.e., observers and parameter estimation.

**3.2. State observers.** In the case of the application of state observers of the Luenberger type [10,23] (the deterministic case) or Kalman filters [24–27] (the stochastic case), residuals are generated using the estimate of system outputs. The estimated outputs are compared with the measured outputs and the residual is equal to the difference between them (Fig. 3). In the stochastic case applying the Kalman filter, the innovation process is used as a residual [25,28]. The application of observer approach for FDI purposes differs from the standard observer applications in control, where the state vector is estimated [24]. Here only output estimation is required. An effective application of observers in control and FDI systems [24,25,29] needs accurate models, which is very diffult to satisfy in practice (model uncertainty). To solve this problem, the theory of observers and filters with unknown inputs for the linear and non-linear dynamical system has been developed [7,9].

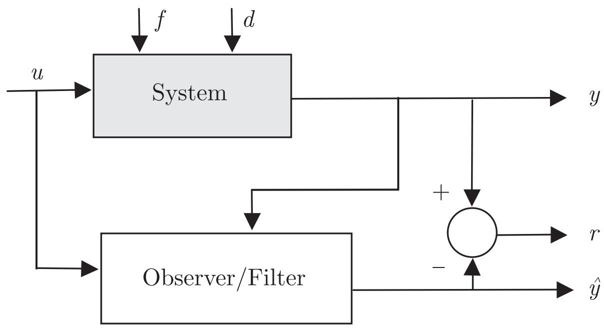


Fig. 3. Observer-based residual generator

**Unknown input observers.** Consider the linear, discrete system described by the state Eqs. [7,9]:

$$\boldsymbol{x}(k+1) = \boldsymbol{A}(k)\boldsymbol{x}(k) + \boldsymbol{B}(k)\boldsymbol{u}(k) + \boldsymbol{Q}\boldsymbol{d}(k) + \boldsymbol{R}_1(k)\boldsymbol{f}(k), \tag{11}$$

$$\boldsymbol{y}(k+1) = \boldsymbol{C}(k)\boldsymbol{x}(k+1) + \boldsymbol{R}_2(k)\boldsymbol{f}(k+1), \tag{12}$$

where $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{Q}$ and $\boldsymbol{C}$ are known matrices, and the other notations are the same as in Eqs. (4)–(5). The term $\boldsymbol{Q}\boldsymbol{d}(k)$ represents model uncertainty as well as disturbances acting on the system.

The design problem of an unknown input observer consists in ensuring asymptotic convergence of the estimation error independly of the unknown input $\boldsymbol{d}(k)$. The general structure of an UIO can be given as [7,9,26]:

$$\boldsymbol{z}(k+1) = \boldsymbol{F}(k+1)\boldsymbol{z}(k) + \boldsymbol{T}(k+1)\boldsymbol{B}(k)\boldsymbol{u}(k) + \boldsymbol{K}(k+1)\boldsymbol{y}(k), \tag{13}$$

$$\hat{\boldsymbol{x}}(k+1) = \boldsymbol{z}(k) + \boldsymbol{H}(k+1)\boldsymbol{y}(k+1), \tag{14}$$

where $\hat{\boldsymbol{x}}(k) \in \mathbb{R}^n$ is the state estimate vector, $\boldsymbol{z} \in \mathbb{R}^n$ is the state vector of the full rank observer, $\boldsymbol{F}$, $\boldsymbol{T}$, $\boldsymbol{K}$ and $\boldsymbol{H}$ are matrices which should satisfy some condition for decoupling the residual from the unknown input $\boldsymbol{d}(k)$.

Taking into account some assumptions, to decouple the unknown input from the state estimation error the following relation should be satisfied [23]:

$$\big(I - \boldsymbol{H}(k+1)\boldsymbol{C}(k+1)\big)\boldsymbol{Q}(k) = 0. \tag{15}$$

The necessary condition for the existence of the solution (15) is $\operatorname{rank}(\boldsymbol{Q}(k)) = \operatorname{rank}(\boldsymbol{C}(k)\boldsymbol{Q}(k))$, and a special solution is given by:

$$\boldsymbol{H}^*(k+1) = \boldsymbol{Q}(k)\Big[\big(\boldsymbol{C}(k+1)\boldsymbol{Q}(k)\big)^T \times \boldsymbol{C}(k+1)\boldsymbol{Q}(k)\Big]^{-1} \times \big(\boldsymbol{C}(k+1)\boldsymbol{Q}(k)\big)^T. \tag{16}$$

The unknown matrix $\boldsymbol{K}_1(k+1)$ can be defined in the same way as for the standard observer [10]. Finally, the state estimation error $\boldsymbol{e}(k+1)$ and the residual $\boldsymbol{r}(k+1)$ are described by the relations

$$\begin{aligned}\boldsymbol{e}(k+1) = \boldsymbol{F}(k+1)\boldsymbol{e}(k) + \boldsymbol{T}(k+1)\boldsymbol{R}_1(k)\boldsymbol{f}(k)\\ - \boldsymbol{H}(k+1)\boldsymbol{R}_2(k+1)\boldsymbol{f}(k+1)\\ - \boldsymbol{K}_1(k+1)\boldsymbol{R}_2\boldsymbol{f}(k),\end{aligned} \tag{17}$$

$$\boldsymbol{r}(k+1) = \boldsymbol{C}(k+1)\boldsymbol{e}(k+1) + \boldsymbol{R}_2(k+1)\boldsymbol{f}(k+1), \tag{18}$$

where $\boldsymbol{r}(k+1) = \boldsymbol{y}(k+1) - \hat{\boldsymbol{y}}(k+1)$ is the residual which depends only on the faults $\boldsymbol{f}(k)$, and is independent of the unknown inputs $\boldsymbol{d}(k)$.

In a pretty simple way the design problem of the Kalman filter with an unknown input for a linear system can be solved [9]. Moreover, for the non-linear case the problem is how to define an accurated model. The difficulty of system modelling consists in the necessity of approximating both the structure and the parameters. To overcome this problem, a modified genetic programing approach [30] for model structure selection combined with the classical techniques for parameter estimation was proposed in [31].

**3.3. Parameter estimation.** Another approach to model-based generation of residuals is based on parameter estimation techniques [5,32]. The structure of the parameter estimation-based fault detection system is shown in Fig. 4, where $\boldsymbol{r}(k) = \boldsymbol{\theta}(k) - \hat{\boldsymbol{\theta}}(k)$, and $k = 1, 2,$ is a discrete time.

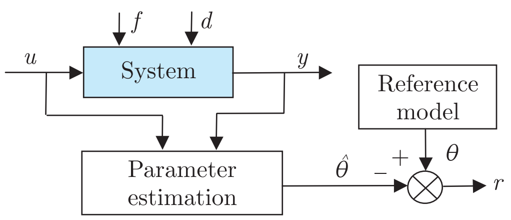


Fig. 4. Scheme of the parameter estimation-based fault detection system

The task consists in detecting faults in a system by measuring the input $\boldsymbol{u}_k$ and the output $\boldsymbol{y}_k$, and then estimating the parameters of the model of the system [31]. In discrete time non-linear input/output dynamics can be generally described by:

$$y(k) = \boldsymbol{g}\left(\boldsymbol{\phi}(k), \boldsymbol{\theta}(k)\right), \tag{19}$$

where $\boldsymbol{\phi}(k)$ may contain the previous or current system input $\boldsymbol{u}(k)$, the previous system or model output $(y(k), \hat{y}(k))$, and the previous prediction error. The model (19) can also be expressed in the state-space form; however, this does not change the general framework. If a fault now occurs in the system, this causes a change $\Delta\boldsymbol{\theta}(k)$ (residual) in the parameter vector $\boldsymbol{\theta}(k)$. Such a residual can then be used to detect and isolate faults.

Model parameters should have physical meaning, i.e., they should correspond to the parameters of the system and if this is not the case the approach is severely limited. When model parameters replicate those of their physical counterparts, the detection and isolation of faults is very straightforward. In a practical situation it can be difficult to distinguish a fault from a change in the parameter vector $\boldsymbol{\theta}(k)$ resulting from time-varying properties of the system. Moreover, the process of fault isolation may become extremely diflicult because model parameters do not uniquely correspond to those of the system. Apart from the above-mentioned difficulties, there are many classes of systems for which it is possible to derive models whose parameters have physical meaning. Distributed parameter systems [33] constitute such an important class. In order to increase the accuracy of parameter estimation and, consequently, the reliability of fault diagnosis, Uciński [33] proposed and developed various procecdures that can be utilised for the development of an experimental design that facilitates high-accuracy parameter estimation.

It should also be pointed out that the detection of faults in sensors and actuators is possible but rather complicated [5,32] with the parameter estimation approach. Robustness with respect to model uncertainty can be tackled relatively easily (especially for linear systems) by employing robust parameter estimation techniques, e.g., the bounded-error approach [20].

## 4. Uncertainty of soft computing models

A common disadvantage of analytical approaches to the FDI system is the fact that a precise mathematical model of the diagnosed plant is required. An alternative solution can be obtained using soft computing techniques, i.e., artificial neural networks, fuzzy logic, expert systems and evolutionary algorithms [14,15,34,35] or their combination as neuro-fuzzy networks [17,18,36,37]. To apply soft computing modelling, empirical data, principles and rules which describe the diagnosed process and other accessible qualitative and quantitative knowledge are required [38].

Below we focus on the problem of designing GMDH networks and Takagi-Sugeno NF systems as well as describing their uncertaity. Knowing such soft computing models' structure and possessing the knowledge regarding their uncertainty it is possible to design robust fault detection schemes by defining adaptive thresholds.

**4.1. Dynamic GMDH networks and their uncertainty.** A disadvantage of most known neural networks is the fact that their architecture is arbitrarily defined [16,39,40,41]. An alternative approach is based on the integration of process training with the choice of the network optimal architecture. Such a design procedure can be obtained by the group method of data handling [39,42,43,44].

The idea of the GMDH is based on replacing the complex model of the process with partial models (neurons) by using the rules of variable selection. As usual, partial models have a small number of inputs $u_i(k)$, $i = 1, 2, \ldots, m$, and are implemented by GMDH neurons. The synthesis process of the GMDH network [44] is based on iterative processing of a sequence of operations. This process leads to the evolution of the resulting model structure in such a way so as to obtain an approximation of the optimal degree of model complexity.

The resulting GMDH neural network is constructed through the connection of a given number of neurons, as shown in Fig. 5.

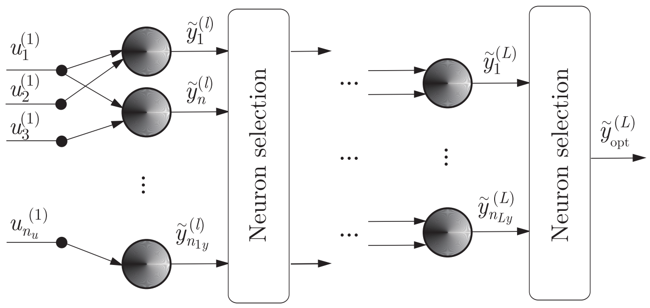


Fig. 5. Principle of the GMDH algorithm

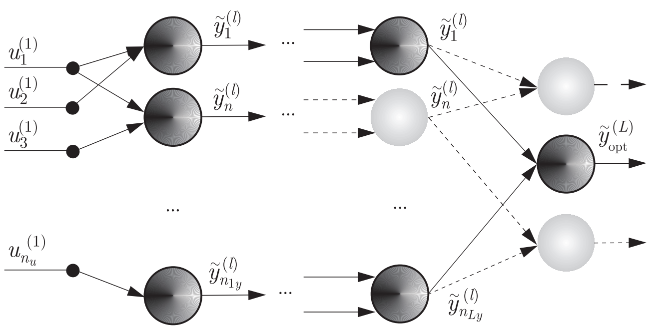

Fig. 6. Final structure of the GMDH neural network

The GMDH neuron has the following structure:

$$\tilde{y}_n^{(l)}(k) = \xi\left(\left(\boldsymbol{z}_n^{(l)}(k)\right)^T \boldsymbol{\theta}_n^{(l)}\right), \tag{20}$$

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 $\boldsymbol{u}(k) \in \mathbb{R}^{n_u}$ of the system, whilst $\xi(\cdot)$ denotes a non-linear invertible activation function, i.e., there exists $\xi^{-1}(\cdot)$. Moreover, $\boldsymbol{z}_n^{(l)}(k) = \boldsymbol{f}\left([u_i^{(l)}(k), u_j^{(l)}(k)]^T\right)$, $i, j = 1, \ldots, n_u$ and $\boldsymbol{\theta}_n^{(l)} \in \mathbb{R}^{n_p}$ are the regressor and the parameter vectors, respectively, and $\boldsymbol{f}(\cdot)$ is an arbitrary bivariate vector function, e.g., $\boldsymbol{f}(\boldsymbol{x}) = [x_1^2, x_2^2, x_1x_2, x_1, x_2, 1]^T$ which corresponds to the bivariate polynomial of the second degree.

An outline of the GMDH algorithm can be as follows [44, 45]:

Step 1: Determine all neurons in the first layer (estimate their parameter vectors $\boldsymbol{\theta}_n^{(l)}$ with the training data set $\mathcal{T}$) whose inputs consist of all possible couples of input variables, i.e., $(n_u - 1)n_u/2$ couples (neurons).

Step 2: Using a validation data set $\mathcal{V}$, not employed during the parameter estimation phase, select several neurons which are best-fitted in terms of the chosen criterion.

Step 3: If the termination condition is fulfilled (the network fits the data with desired accuracy or the introduction of new neurons did not induce a significant increase in approximation abilities of the neural network), then Stop, otherwise use the outputs of the best-fitted neurons (selected in Step 2) to form the input vector for the next layer, and then go to Step 1.

To obtain the final structure of the network (Fig. 6), 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 GMDH neural network. The feature of the above algorithm is that the techniques for parameter estimation of linear-in-parameter models can be used during the realisation of Step 1. Indeed, since $\xi(\cdot)$ is invertible, the neuron (20) can relatively easily be transformed into a linear-in-parameter one.

The objective of structure and parameter identification is to obtain a mathematical description of a real system based on input-output measurements. Irrespective of the identification method used, there is always the problem of model uncertainty, i.e., the model-reality mismatch. Even though the application of the GMDH approach to model structure selection can improve the quality of the model, the resulting structure is not the same as that of the system. It can be shown [45] that the application of the classical evaluation criteria like the Akaike Information Criterion and the Final Prediction Error [46] can lead to the selection of inappropriate neurons and, consequently, to unnecessary structural errors.

Apart from the model structure selection stage, inaccuracy in parameter estimates also contributes to modelling uncertainty. Indeed, while applying the least-square method to parameter estimation of neurons (20), a set of restrictive assumptions has to be satisfied. The first, and the most controversial, assumption is that the structure of the neuron is the same as that of the system (no structural errors). In case of the GMDH neural network, this condition is extremely difficult to satisfy. Indeed, neurons are created based on two input variables selected from $\boldsymbol{u}$ and hence it is impossible to eliminate the structural error. Another assumption concerns transformation with $\xi^{-1}(\cdot)$. The usual statistical parameter estimation methods, i.e., the least-square method or other methods considered in [21,47] assume that data are corrupted by errors which can be modelled as realisations of independent random variables, with a known or parameterised distribution. A more realistic approach is to assume that errors lie between given prior bounds.

Let us consider the following system:

$$y(k) = \left(\boldsymbol{z}_n^{(l)}(k)\right)^T \boldsymbol{\theta}_n^{(l)} + \varepsilon_n^{(l)}(k). \tag{21}$$

The problem is to obtain the parameter estimate vector $\hat{\boldsymbol{\theta}}_n^{(l)}(k)$, as well as associated parameter uncertainty required to design a robust fault detection system. 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:

$$\tilde{y}^m(k) \le y(k) \le \tilde{y}^M(k), \tag{22}$$

where $\tilde{y}^m(k)$ and $\tilde{y}^M(k)$ are the minimum and maximum admissible values of the model output that are consistent with the input-output measurements of the system.

Moreover, it is assumed that $\varepsilon(k)$ is bounded as follows:

$$\varepsilon^m(k) \le \varepsilon(k) \le \varepsilon^M(k), \tag{23}$$

where the bounds $\varepsilon^m(k)$ and $\varepsilon^M(k)$ ($\varepsilon^m(k) < \varepsilon^M(k)$) are known *a priori*. The idea underlying the bounded-error approach is to obtain a feasible parameter set [45]. This set can be defined as

$$\begin{aligned}\mathbb{P} = \big\{\boldsymbol{\theta} \in \mathbb{R}^{n_\theta} \mid y(k) - \varepsilon^M(k) \le \boldsymbol{z}^T(k)\boldsymbol{\theta} \\ \le y(k) - \varepsilon^m(k)\,, k = 1,\ldots,n_{\mathcal{T}}\big\}\,,\end{aligned} \tag{24}$$

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:

$$\mathbb{P} = \bigcap_{k}^{n_{\mathcal{T}}} \mathbb{S}(k), \tag{25}$$

where each pair defines the parameter strip:

$$\begin{aligned}\mathbb{S}(k) = \big\{\boldsymbol{\theta} \in \mathbb{R}^{n_\theta} \mid y(k) - \varepsilon^M(k) \le \boldsymbol{z}^T(k)\boldsymbol{\theta} \\ \le y(k) - \varepsilon^m(k)\big\}.\end{aligned} \tag{26}$$

Any parameter vector contained in $\mathbb{P}$ is a valid estimate of $\boldsymbol{\theta}$. In practice, the centre (in some geometrical sense) of $\mathbb{P}$ is chosen as the parameter estimate $\hat{\boldsymbol{\theta}}$, e.g.,

$$\hat{\theta}_i = \frac{\theta_i^m + \theta_i^M}{2}, \quad i = 1,\ldots,n_\theta, \tag{27}$$

where

$$\theta_i^m = \arg\min_{\theta\in\mathbb{P}} \theta_i, \quad i = 1,\ldots,n_\theta, \tag{28}$$

$$\theta_i^M = \arg\max_{\theta\in\mathbb{P}} \theta_i, \quad i = 1,\ldots,n_\theta. \tag{29}$$

The problem (28) and (29) can be solved with the well-known linear programming techniques [20,48], but when $n_{\mathcal{T}}$ and/or $n_\theta$ are large, the computational cost may be significant. This constitutes the main drawback of the approach. In our case, GMDH neurons represent a relatively small dimension and therefore the BEA can be employed. On the other hand, the main difficulty associated with the BEA concerns *a priori* knowledge regarding the error bounds $\varepsilon^m(k)$ and $\varepsilon^M(k)$. However, these bounds can also be estimated [20] by assuming that $\varepsilon^m(k) = \varepsilon^m$, $\varepsilon^M(k) = \varepsilon^M$, and then suitably extending the unknown parameter vector $\boldsymbol{\theta}$ by $\varepsilon^m$ and $\varepsilon^M$. Determining the bounds can now be formulated as follows:

$$(\varepsilon^m, \varepsilon^M) = \arg\min_{\varepsilon^M \ge 0,\ \varepsilon^m \le 0} (\varepsilon^M - \varepsilon^m), \tag{30}$$

with respect to the following constraints:

$$y(k) - \varepsilon^M \le \boldsymbol{z}^T(k)\boldsymbol{\theta} \le y(k) - \varepsilon^m\,, k = 1,\ldots,n_{\mathcal{T}}. \tag{31}$$

To solve the problem (30), the well-known simplex method along with the strategy presented in [49] can be applied.

Using the methodology described above it is possible to obtain the parameter estimate $\hat{\boldsymbol{\theta}}$ and the associated feasible parameter set $\mathbb{P}$. However, from the practical point of view, it is more convenient to obtain model output uncertainty, i.e., the interval in which the "true" model output $\tilde{y}(k)$ can be found. This kind of knowledge makes it possible to obtain an adaptive threshold [10], and hence to develop a fault diagnosis scheme that is robust to model uncertainty.

If there is no error in the regressor, then the problem of determining model output uncertainty can be solved as follows:

$$\boldsymbol{z}^T(k)\boldsymbol{\theta}^m(k) \le \boldsymbol{z}^T(k)\boldsymbol{\theta} \le \boldsymbol{z}^T(k)\boldsymbol{\theta}^M(k), \tag{32}$$

where

$$\boldsymbol{\theta}^m(k) = \arg\min_{\boldsymbol{\theta}\in\mathbb{V}} \boldsymbol{z}^T(k)\boldsymbol{\theta}, \tag{33}$$

$$\boldsymbol{\theta}^M(k) = \arg\max_{\boldsymbol{\theta}\in\mathbb{V}} \boldsymbol{z}^T(k)\boldsymbol{\theta}. \tag{34}$$

The computation of (33) and (34) is realised by multiplying parameter vectors corresponding to all vertices belonging to $\mathbb{V}$ by $\boldsymbol{z}^T(k)$. Since (32) describes neuron output uncertainty, the system output will satisfy:

$$\boldsymbol{z}^T(k)\boldsymbol{\theta}^m(k) + \varepsilon^m(k) \le y(k) \le \boldsymbol{z}^T(k)\boldsymbol{\theta}^M(k) + \varepsilon^M(k). \tag{35}$$

The neuron output uncertainty defined by (32) and the corresponding system output uncertainty (35) are presented in Figs. 7 and 8, respectively. According to the GMDH network structure, the neurons in the $l$-th ($l > 1$) layer are fed with the outputs of the neurons from the $(l-1)$-th layer. Since (32) describes model output uncertainty, the parameters of the neurons in layers have to be obtained with an approach that solves the problem of an uncertain regressor [20].

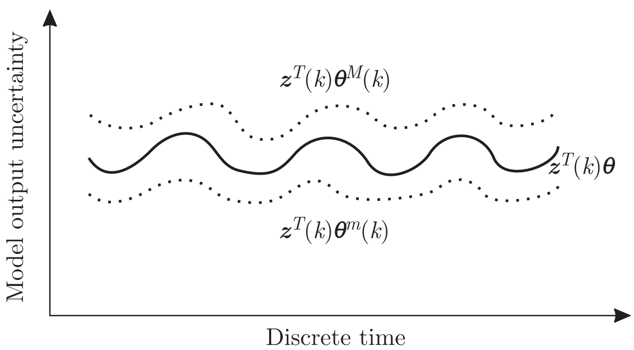


Fig. 7. Model output uncertainty for the error-free regressor

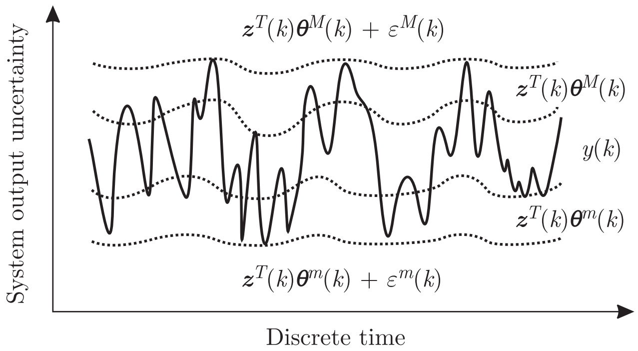


Fig. 8. System output uncertainty for the error-free regressor

Taking into account an error in the regressor, it was shown [50,51] that model output uncertainty has the following form:

$$\tilde{y}^m(k)\left(\theta'^m(k),\theta''^m(k)\right) \le z_n^T\theta \le \tilde{y}^M(k)\left(\theta'^M(k),\theta''^M(k)\right), \tag{36}$$

where

$$\tilde{y}^m(k)\left(\theta'^m(k),\theta''^m(k)\right) = \left(z(k)-e^M(k)\right)^T\theta'^m(k) + \left(e^m(k)-z(k)\right)^T\theta''^m(k), \tag{37}$$

$$\tilde{y}^M(k)\left(\theta'^M(k),\theta''^M(k)\right) = \left(z(k)-e^m(k)\right)^T\theta'^M(k) + \left(e^M(k)-z(k)\right)^T\theta''^M(k), \tag{38}$$

and

$$\left(\theta'^m(k),\theta''^m(k)\right) = \arg\min_{(\theta',\theta'')\in V}\tilde{y}^m(k)\left(\theta',\theta''(k)\right), \tag{39}$$

$$\left(\theta'^M(k),\theta''^M(k)\right) = \arg\max_{(\theta',\theta'')\in V}\tilde{y}^M(k)\left(\theta',\theta''(k)\right). \tag{40}$$

Moreover, $e(k)$ denotes the error in the regressor $z(k)$, and it was assumed that each parameter $\theta_i$ can be replaced by $\theta_i = \theta_i' - \theta_i''$, $\theta_i', \theta_i'' \ge 0$, $i = 1, \ldots, n_\theta$.

Using (36) it is possible to obtain system output uncertainty:

$$\tilde{y}^m(k)\left(\theta'^m(k),\theta''^m(k)\right) + \varepsilon^m(k) \le y(k) \le \tilde{y}^M(k) \times \left(\theta'^M(k),\theta''^M(k)\right) + \varepsilon^M(k). \tag{41}$$

The presented bounded-error approach has been applied to the synthesis of GMDH networks [45].

**4.2. Takagi-Sugeno neuro-fuzzy networks.** For the last few years an increasing number of authors [15,18,36,37,52] have been using integrated neural-fuzzy models in order to benefit from advantages of both. The neuro-fuzzy model combines, in a single framework, both numerical and symbolic knowledge. Automatic linguistic rule extraction is a useful aspect of NF, especially when little or no prior knowledge about the process is available [36,52]. For example, an NF model of a non-linear dynamical system can be identified using empirical data. This model may give us some insight about the non-linearity and dynamical properties of the system [34]. From the mentioned literature it follows that NF-based FDI is advantageous when [38]:

— enough certain system information is not available,
— physical or semi-physical models are difficult to obtain,
— sensor measurements are incomplete, assumed or missing,
— training data are difficult to obtain because of a lack of control of some input variable,
— the system exhibits strong non-linear static and dynamic behaviour,
— during the normal operation, frequency components of the input signal are not complete enough to build a model that can be reliable for all possible frequencies.

Two types of NF networks are commonly used for the modelling purpose: the Mamdani NF network and the Takagi-Sugeno NF network. Generally, Takagi-Sugeno structures express better performance in modelling than other structures due to their possibility to decompose non-linear systems into a collection of local linear models, and therefore the paper concentrates on such structures. The main problem which arises during designing Takagi-Sugeno networks is the question about a suitable number of rules which ensure modelling accuracy.

This is usually a trade-off between the complexity of the network and its accuracy. The existing methods for determining the structure of NF network are time consuming (genetic algorithms [53,54], clustering algorithms [55], partitioning algorithms [56]) and do not assure the accuracy of the designed model. Below an effective method [57,58] for structure determination based on the bounded-error approach will be considered.

The structure of the Takagi-Sugeno system could be presented in the form of a layered topology similar to the neural network [36,37]. However, knowledge coded in this structure could be viewed in the form of fuzzy rules [59,60]:

$$R_i: \textit{ IF } x \textit{ is } A_i \textit{ THEN } y_i = z_i^T\theta_i, \tag{42}$$

where $x$ is the vector of global network inputs, $A_i$ is the multivariate fuzzy set, $y_i$ is the scalar output of the rule, $z_i$ is the vector of local linear system inputs, $\theta_i$ is the vector of local linear system parameters, and $i$ is the index of the rule. Fuzzy sets usually have Gaussian membership functions. The global output of the NF network is a composition of the responses of all rules:

$$y = \frac{\sum_{i=1}^{n}\mu_k y_i}{\sum_{i=1}^{n}\mu_i}, \tag{43}$$

where $y$ is the global output of the network, $\mu_i$ is the membership degree achieved for $i$-th rule, $y_i$ is the output of the $i$-th rule (local linear system), $n$ is the number of rules. It is worth to noticing that the number of rules determines the number of local linear models, which are responsible for piecewise local linear approximation of the non-linear system. It could be shown that the number of rules has a strong influence on the accuracy of the global model and its complexity [57,61].

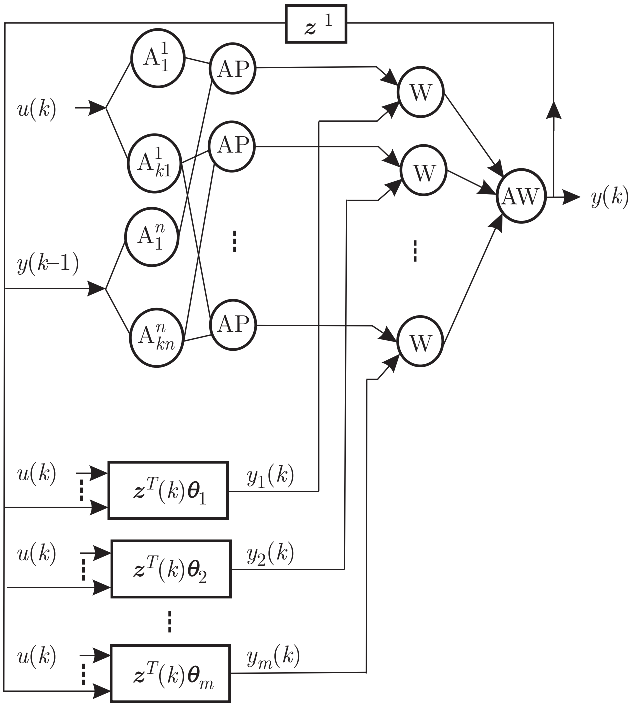


Fig. 9. Sample dynamic Takagi-Sugeno network, where AP is logical conjunction, W – inference operation, and AW – conclusion aggregation

It is very important to include dynamics [61,62] in the NF network because real processes are usually dynamic. It could be done by introducing into the input vector $\boldsymbol{z}_i$ the delayed inputs $\boldsymbol{u}_i$ of the local model and the delayed output of the local output $y_i$, i.e., $\boldsymbol{z}_i = [u_i(k), u_i(k-1), \ldots, u_i(k-n_a), y_i(k-1), y_i(k-2), \ldots, y_i(k-n_b)]$. The sample layered structure of the dynamic Takagi-Sugeno network is presented in Fig. 9.

To apply the bounded-error approach for determining the uncertainty of the NF network considered, we need to assume that such network is linear in parameters [20]. In this case a model uncertainty will be characterized by a feasible set of parameters defined by the BEA.

Let us consider the following Takagi-Sugeno NF model:

$$\tilde{y}(k) = \sum_{i=1}^{n} \phi_i(k)\tilde{y}_i(k), \tag{44}$$

where, $\tilde{y}_i(k)$ is the output of the $i$-th rule and

$$\phi_i(k) = \frac{\mu_i(k)}{\sum_{j=1}^{n} \mu_j(k)}. \tag{45}$$

The model described by Eq. (44) could be viewed as a system linear in parameters:

$$\tilde{y} = \boldsymbol{x}^T(k)\boldsymbol{\theta}, \tag{46}$$

where

$$\boldsymbol{x}(k) = \begin{bmatrix} \phi_1(k)\boldsymbol{z}_1(k) \\ \phi_2(k)\boldsymbol{z}_2(k) \\ \vdots \\ \phi_n(k)\boldsymbol{z}_n(k) \end{bmatrix}, \quad \boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\theta}_1 \\ \boldsymbol{\theta}_2 \\ \vdots \\ \boldsymbol{\theta}_n \end{bmatrix},$$

if the parameters of fuzzy sets are treated like constant values. Let us define the output error $\varepsilon(k)$:

$$\varepsilon(k) = y(k) - \boldsymbol{x}^T(k)\boldsymbol{\theta}, \tag{47}$$

where $y(k)$ is the output of the system. The usual statistical parameter estimation approaches assume that data are corrupted by errors which can be modelled as realizations of independent random variables, with a known or parameterized distribution [56]. The bounded-error approach is more realistic because it assumes that errors lie between given priori bounds [11, 20]:

$$\varepsilon^m(k) \leq \varepsilon(k) \leq \varepsilon^M(k). \tag{48}$$

Let us assume that

$$\varepsilon^M(k) = \varepsilon,\ \varepsilon^m(k) = -\varepsilon. \tag{49}$$

Thus the feasible set of parameters for $N$ data points is:

$$\mathbb{P} = \left\{\boldsymbol{\theta} \in \mathbb{R}^{n_\theta} \mid y(k) + \varepsilon \leq \boldsymbol{x}^T(k)\,\boldsymbol{\theta} \leq y(k) - \varepsilon;\ k = 1, \ldots, N\right\}. \tag{50}$$

Then the confidence interval for the system output is given by:

$$\boldsymbol{x}^T(k)\,\boldsymbol{\theta}^m(k)\,\varepsilon \leq y(k) \leq \boldsymbol{x}^T(k)\,\boldsymbol{\theta}^M(k) + \varepsilon, \tag{51}$$

where

$$\boldsymbol{\theta}^M(k) = \arg\max_{\boldsymbol{\theta}\in P} \boldsymbol{x}^T(k)\,\boldsymbol{\theta}, \tag{52}$$

$$\boldsymbol{\theta}^m(k) = \arg\min_{\boldsymbol{\theta}\in P} \boldsymbol{x}^T(k)\,\boldsymbol{\theta}. \tag{53}$$

This algorithm requires to determine the set of all vertices $\mathbb{W}$ of the convex polyhedron $\mathbb{P}$. This process is so time consuming that it is hard to employ the described algorithm for models with more then 6 parameters. Fortunately, the recursive Outer-Bounding Ellipsoid (OBE) algorithm [9,20] is able to approximate the area $\mathbb{P}$ by enclosing it by the ellipsoid $\mathbb{E}$ and is not so time consuming [21]. The data in this algorithm are taken into account one after another to construct a succession of ellipsoids containing all values parameters consistent with all previous measurements.

Using the OBE algorithm, the confidence interval for the system output can be given by [57,61]:

$$\boldsymbol{x}^T(k)\hat{\boldsymbol{\theta}} - \sqrt{\boldsymbol{x}^T(k)\boldsymbol{M}\boldsymbol{x}(k)} - \varepsilon \leq y(k) \leq \boldsymbol{x}^T(k)\hat{\boldsymbol{\theta}} + \sqrt{\boldsymbol{x}^T(k)\boldsymbol{M}\boldsymbol{x}(k)} + \varepsilon, \tag{54}$$

where $\hat{\boldsymbol{\theta}}$ is the center of the ellipsoid, $\boldsymbol{M}$ is the positive define matrix, which specifies the size and orientation of the ellipsoid.

## 5. Robust fault detection systems

As was mentioned, the ideal of model-based FDI is to generate signals that reflect inconsistences between the nominal and faulty system operation. On the other hand, the uncertainty of the model could dramatically decrease the reliability of fault detection if it is not taken into consideration. Two main approaches have been proposed to overcome the described problem [1,7]: the active approach, which is usually based on robust observers, and the passive approach, which is usually based on the adaptive threshold computed for the residual.

In the threshold test, the residual $\boldsymbol{r}(k) = y(k) - \tilde{y}(k)$ has to be checked against thresholds, which traditionally are constant. In this case, the decision is significantly affected by model uncertainty. If the threshold chosen is too small, uncertainties cause false alarms, if opposite – small faults cannot be detected. A more effective solution can be obtained if the threshold is adapted to the time evolution of the residual in the fault-free case [12]. The idea of threshold adaptation is shown in Fig. 10. It is clear that if a fixed threshold (the dashed lines) is used, a false alarm occurs at the time $T_{fa}$, and, on the other hand, the fault at $T_f$ cannot be detected. If an adaptive threshold is used (the dashed lines) which follows in some way the residual caused by the input in the fault-free case, both the false alarm can be avoided and the fault at $T_f$ be detected. In a simple way the shape of the adaptive threshold can be found empirically by the inspection of the shape of the residual under fault-free operation [34] or using the analytical approach [8].

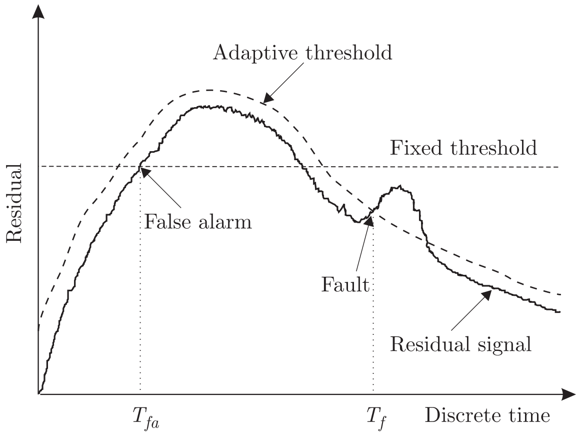


Fig. 10. Illustration of the concept of the adaptive threshold

**5.1. Adaptive threshold with the GMDH model.** To show how to develop an adaptive threshold with the GMDH model, the confidence range of the output system (41) is considered. Since the residual is given by [4,5]:

$$r(k) = y(k) - \tilde{y}(k), \tag{55}$$

then as a result of substituting (55) into (41), an adaptive threshold can be described as [50]:

$$\tilde{y}^m(k)\left(\boldsymbol{\theta}'^m(k), \boldsymbol{\theta}''^m(k)\right) - \tilde{y}(k) + \varepsilon^m(k) \le r(k) \\ \le \tilde{y}^M(k)\left(\boldsymbol{\theta}'^M(k), \boldsymbol{\theta}''^M(k)\right) - \tilde{y}(k) + \varepsilon^M(k). \tag{56}$$

The principle of fault detection with the developed adaptive threshold is shown in Fig. 11. At the time $T_f$ the residual crosses the upper bound of the confidence range of the adaptive threshold, and this moment is indicated as fault appearance.

**5.2. Adaptive threshold with the NF model.** Using the same approach as for the GMDH network, an adaptive threshold with the Takagi-Sugeno network can be directly defined by applying the confidence range of the output system (54). After substituting (55) into (54), the adaptive threshold can be put in the following form [61]:

$$\boldsymbol{x}^T(k)\hat{\boldsymbol{\theta}} - \sqrt{\boldsymbol{x}^T(k)\boldsymbol{M}\boldsymbol{x}(k)} - \varepsilon - \tilde{y}(k) \le r(k) \\ \le \boldsymbol{x}^T(k)\hat{\boldsymbol{\theta}} + \sqrt{\boldsymbol{x}^T(k)\boldsymbol{M}\boldsymbol{x}(k)} + \varepsilon - \tilde{y}(k). \tag{57}$$

It has to be pointed out that the presented approach for computing the adaptive threshold for the neuro-fuzzy model assumes that the input vector $\boldsymbol{x}$ is not corrupted by errors. In real situations sometimes this assumption may not be fulfilled.

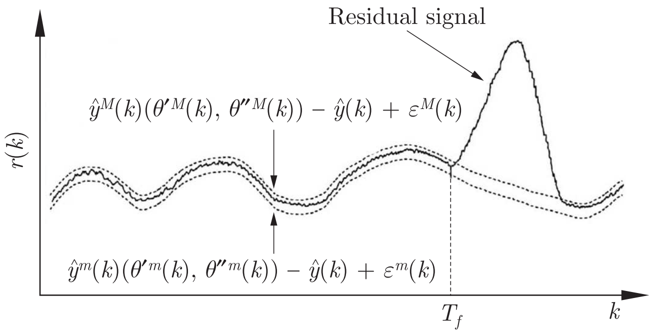


Fig. 11. Adaptive threshold with the uncertainty GMDH model

## 6. Experimental results

During the realization (2001–2004) of the Research Training Network on the *Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems (DAMADICS)*[63], analytical and soft computing approaches to on-line diagnosis of a 5-stage evaporisation plant of the sugar factory in Lublin (Poland) were studied and developed. One of the subtasks of this project was to develop robust fault detection methods using soft computing models with an adaptive decision threshold.

**6.1. Process description.** One of the most important components of each evaporation station is an actuator consisting of three parts: a control value, a pneumatic spring-and-diaphragm actuator and a positioner. Actuators are installed mainly in a harsh environment such as high temperature and presure, low or high humidity, dusty pollutants, chemical solvents, aggressive media, vibrations, etc. These conditions (external and internal) have a crucial influence on the actuator's predicted lifetime and, moreover, cause different malfunctions or failures. To avoid the damage caused by incipient and sudden faults, on-line diagnosis of actuators is needed [64,65].

The scheme of the actuator with an intelligent positioner is given in Fig. 12. The following notations are used: $V_1$, $V_2$ and $V_3$ are cut-off valves, $ACQ$ is a data acquisition unit, $CPU$ is a positioner central processing unit, $E/P$ is an electro-pneumatic transducer, and $DT$, $PT$ and $FT$ denote displacement, pressure and volume flow tranducers, respectively. For remote on-line diagnostics, the following measured variables are accessible: the flow rate of juice after the control valve ($F$), the actuator's rod displacement ($X$), the input set-point ($C_V$), juice temperature at the input of the control valve ($T_1$), and juice pressures at the input and outlet of the control valve, respectively ($P_1$ and $P_2$).

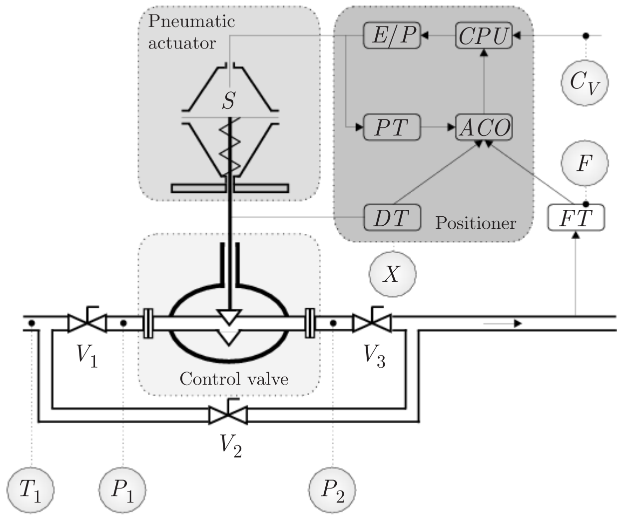


Fig. 12. Scheme of the intelligent actuator

**6.2. GMDH network modelling and detection.** Based on the actuator benchmark definition [63], two structural models can be defined [64]:

$$F = f_F(X, P_1, P_2, T_1), \tag{58}$$

$$X = f_X(C_V, P_1, P_2, T_1), \tag{59}$$

where $f_F(\cdot)$ and $f_X(\cdot)$ denote unknown non-linear functions of the flow rate and displacement, respectively. Using these functional relations, GMDH neural dynamic models were developed. For the research purpose, 19 faults $(f_1, f_2, \ldots, f_{19})$ were selected and grouped in four sets [63]: the faults of the control value, the pneumatic actuator, and the positioner. The general/external faults create the fourth set.

For the purpose of fault detection, two GMDH neural models corresponding to the relations (58) and (59) were built. During the synthesis process of these networks, the so-called selection method was employed [45,66], and the final structures are shown in Fig. 13.

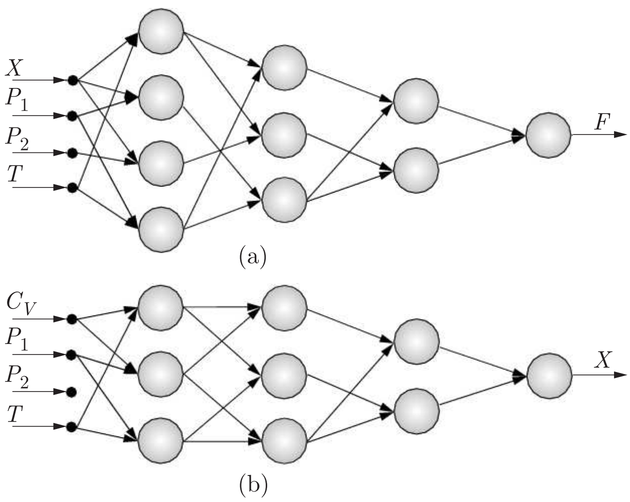


Fig. 13. Final structures of GMDH models: for $F = f_F(\cdot)$ (a) and for $X = f_X(\cdot)$ (b)

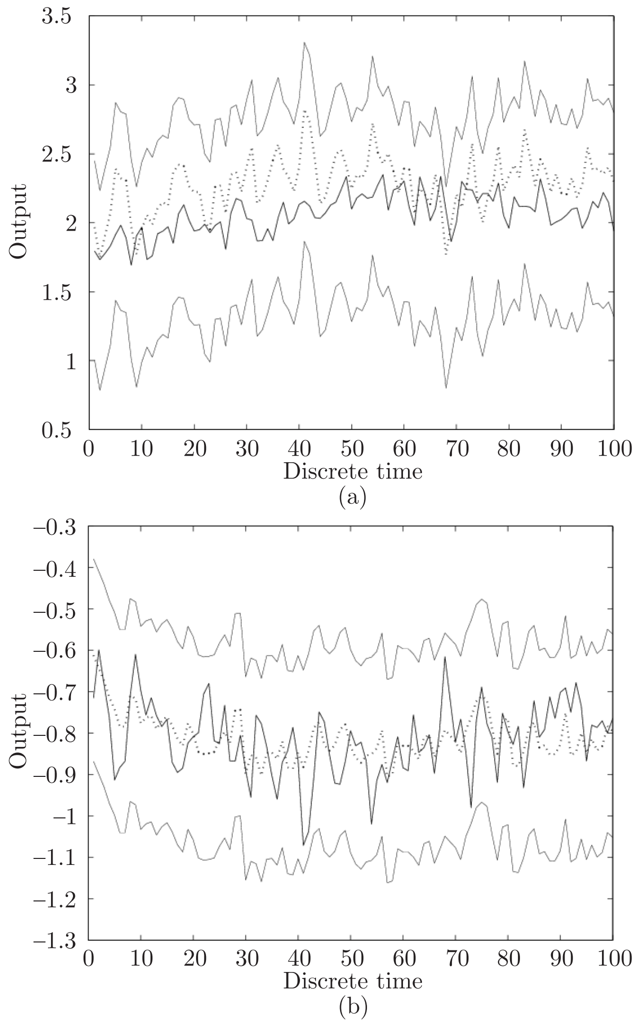


Fig. 14. The model and the system outputs with corresponding uncertainly range

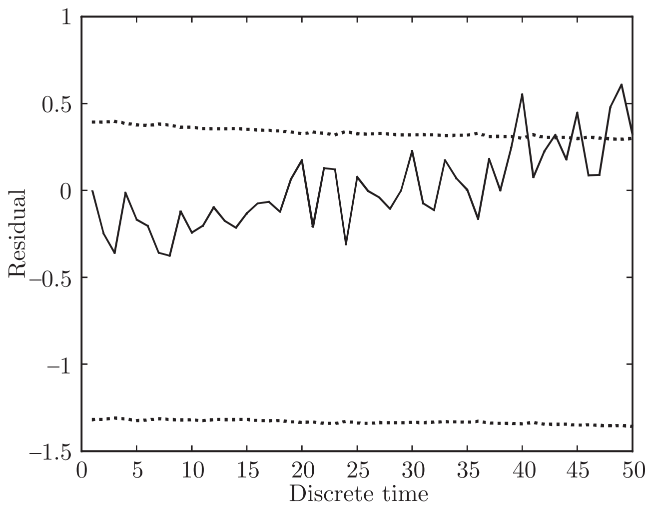


Fig. 15. Residual and adptive threshold for the fault $f_{18}$

Figures 14(a) and 14(b) present the modelling abilities of the obtained models $F = f_F(\cdot)$ and $X = f_X(\cdot)$ as well the corresponding system output uncertainty. The thick solid line represents the real system output, the thin solid lines correspond to system output uncertainty, and the dashed line denotes the model response. From Fig. 14, it is clear that the system response is contained within the system output bounds, which were designed with the estimated output errors bounds [45].

The main objective of this application study was to develop a fault detection scheme for the valve actuator. Employing the models $F = f_F(\cdot)$ and $X = f_X(\cdot)$ for robust fault detection with the approach proposed in [45], the selected results are shown in Figs. 15 and 16. These figures present residuals (the solid lines) and their bounds given by adaptive thresholds (the dashed lines). It is clear that both faults $f_{18}$ and $f_{19}$ are detected.

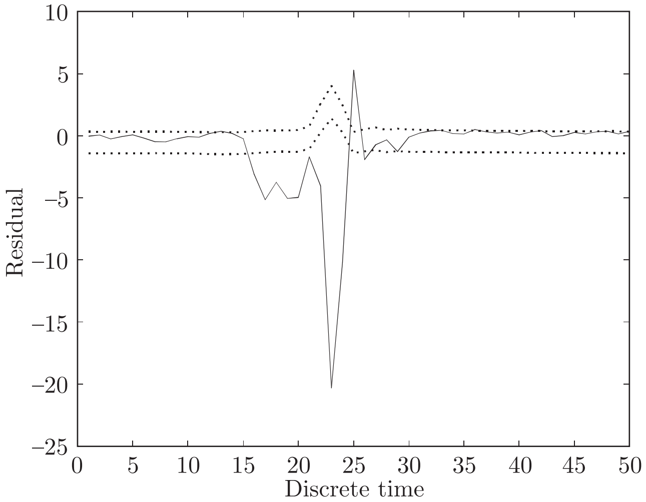


Fig. 16. Residual and adaptive threshold for the fault $f_{19}$

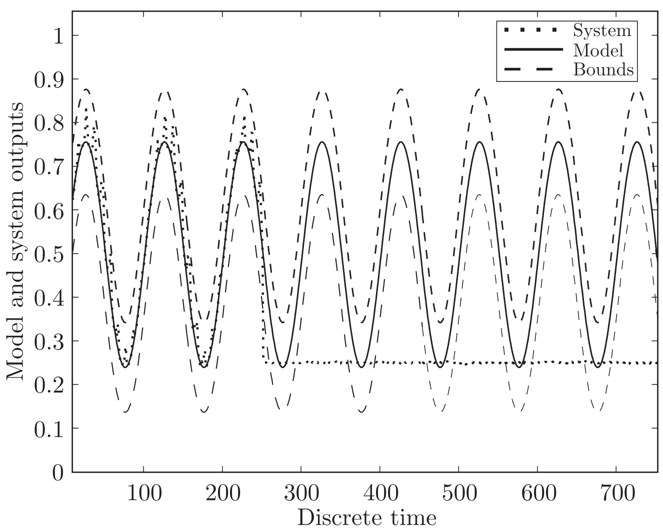


Fig. 17. Model and system output with system output bounds (big fault)

**6.3. Takagi-Sugeno network modelling and detection.** Consider the same actuator benchmark and two structural models given by (57)–(58). Applying the method for structure generation of NF models [57,61] and the results presented in Subsection 4.2, two NF models can be defined. The obtained structures are described in Table 1.

Table 1
Neuro-fuzzy models

| quantity | $f_F$ | $f_X$ |
|---|---|---|
| global inputs | $X$ | $C_V$ |
| local inputs | $X, P_1, P_2, T_1$ | $C_V, P_1, P_2, T_1$ |
| no. of rules | 7 | 3 |

The parametres of fuzzy sets were estimated from the results obtained during structure generation and the parameters of the consequents were estimated using the OBE algorithm.

The first step of the experimental study was to present the modelling abilities of the obtained NF models and, additionaly, their system output uncertainty. Figure 17 presents the modelling abilities of the obtained model along with corresponding system output bounds. At the time $T_f = 250$ the big fault (the valve was blocked) occured.

From Fig. 18, which shows the residual and its bounds given by the adaptive threshold, follows that this fault is detected very fast, with a small delay, approximately 5 units.

The developed fault detection scheme with NF models using the available data containing 44 faulty scenarios generated by the actuar simulator [63] was tested as well. The faults were divided into two main groups: abrupt and incipient faults. Then abrupt faults were divided into three groups: small, medium and big faults.

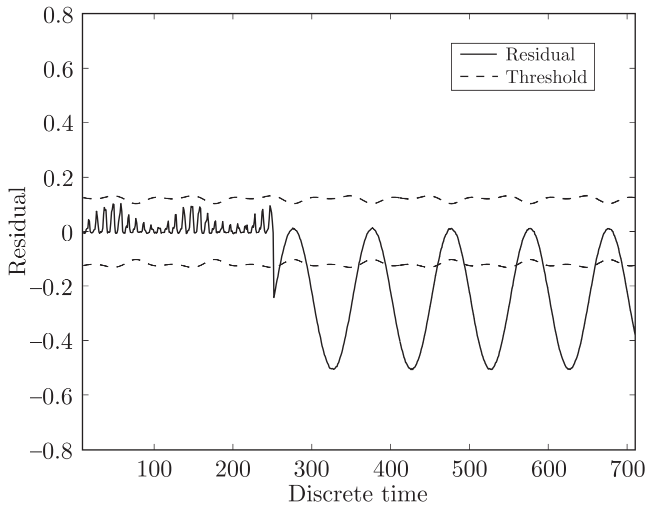


Fig. 18. Residual and adaptive threshold (big fault)

The fault detection results obtained for all scenarios are shown in Table 2, where the following notations are introduced: Y indicates the fault detected using the designed NF models, N indicates the fault that was not detected by the designed NF models.

From Table 2 it follows that most faults can be detected, however, there are a few faults that cannot. The reason for such a situation was that the system output bounds obtained by the OBE algorithm were too large and hence sensitivity to faults was not high enough. This means that it is necessary to employ a more accurate technique than the OBE algorithm.

Table 2
Fault detection results

| No. | Description | S | M | B | I |
|---|---|---|---|---|---|
| | Control valve faults | | | | |
| $f_1$ | Valve clogging | Y | Y | Y | |
| $f_2$ | Sedimentation | | | Y | Y |
| $f_3$ | Seat erosion | | | | Y |
| $f_4$ | Bushing frictions | | | | Y |
| $f_5$ | External leakage | | | | N |
| $f_6$ | Internal leakage | | | | Y |
| $f_7$ | Medium evaporation | Y | Y | Y | |
| | Servo-motor faults | | | | |
| $f_8$ | Twisted piston rod | N | N | Y | |
| $f_9$ | Housing | | | | N |
| $f_{10}$ | Diaphragm perforation | Y | Y | Y | |
| $f_{11}$ | Spring fault | | | Y | Y |
| | Positioner faults | | | | |
| $f_{12}$ | E/P transducer fault | N | N | N | |
| $f_{13}$ | Rod displ. sensor fault | Y | Y | Y | Y |
| $f_{14}$ | Pressure sensor fault | N | N | N | |
| $f_{15}$ | Feedback fault | | | Y | |
| | External faults | | | | |
| $f_{16}$ | Pressure drop | Y | Y | Y | |
| $f_{17}$ | Unexpected pressure change | | | Y | Y |
| $f_{18}$ | Opened bypass valves | Y | Y | Y | Y |
| $f_{19}$ | Flow rate sensor fault $F$ | Y | Y | Y | |

(S – small, M – medium, B – big, I – incipient)

## 7. Conclusions

The main purpose of this paper was to consider a robust model-based fault detection system applying analytical and soft computing models. Special attention was paid to the uncertainty of such models [67] and their usefulness in fault diagnosis. In particular, uncertainties of GMDH neural networks and Takagi-Sugeno NF networks were considered. The proposed approach was based on the bounded-error approach, which is superior to the celebrated least-square method in many practical applications. It was shown that the defined confidence interval for the system output of the GMDH and Takagi–Sugeno networks can be used to develop an adaptive threshold that permits robust fault detection. In the last part, an experimental study performed with the DAMADICS benchmark problem showed the effectiveness of such robust fault detection based on the uncertainty of soft computing models.

## References

[1] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*, New York: Springer-Verlag, 2003.

[2] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*, New York: Marcel Dekker, Inc., 1998.

[3] J. Korbicz, "Advances in fault diagnosis systems", *Proc. 10th IEEE Int. Conf. on Methods and Models in Automation and Robotics MMAR*, 725–733 (2004).

[4] J. Korbicz, J.M. Kościelny, Z. Kowalczuk, and W. Cholewa (eds.), *Fault Diagnosis. Models, Artificial Intelligence, Applications*. Berlin: Springer-Verlag, 2004.

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

[6] J.M. Kościelny, *Diagnostics of Automatized Industrial Processes*, Warsaw: Akademicka Oficyna Wydawnicza, EXIT, 2001, (in Polish).

[7] J. Chen and R.J. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*, London: Kluwer Academic Publishers, 1999.

[8] P.M. Frank, "Handling modelling uncertainty in fault detection and isolation systems", *Proc. 9th Int. Conf. IPMU*, Annecy, France, 1729–1749 (2002).

[9] M. Witczak, *Identification and Fault Detection of Non-linear Dynamic Systems*, University of Zielona Góra Press, 2003.

[10] P.M. Frank, G. Schreier, and E.A. Garcia, "Nonlinear observers for fault detection and isolation", in: *New Directions in Nonlinear Observer Design* (Nijmeijer H., Fossen T.I., eds.), Berlin: Springer-Verlag, 1999.

[11] M. Witczak, A. Obuchowicz, and J. Korbicz, "Genetic programming based approaches to identification and fault diagnosis of nonlinear dynamic systems", *Int. J. Control*, 75 (13), 1012–1031 (2002).

[12] P.M. Frank, "Enhancement of robustness in observer-based fault detection", *Int. J. Control* 59, 955–981 (1994).

[13] J. Korbicz and C. Cempel (eds.), "Analytical and knowledge-based redundancy in fault detection and diagnosis", *Appl. Math. and Comp. Sci., Spec. Issue* 3 (3), (1993).

[14] J.M.F. Calado, J. Korbicz, K. Patan, R.J. Patton, and J.M.G. Sá da Costa, "Soft computing approaches to fault diagnosis for dynamic systems", *European Journal of Control* 7 (2–3), 248–286 (2001).

[15] R.J. Patton and J. Korbicz (eds.), "Advances in computational intelligence for fault diagnosis systems", Special Issue of *Int. J. Appl. Math. and Comp. Sci.* 9 (3), (1999).

[16] J. Korbicz, A. Obuchowicz, and D. Uciński, *Artificial Neural Networks. Foundation and Applications*, Warsaw: Akademicka Oficyna Wydawnicza, PLJ, 1994, (in Polish).

[17] J. Korbicz and M. Kowal, "Neuro-fuzzy systems in process diagnosis", in: *Fuzzy Sets and Their Applications* (J. Chojcan, J. Łęski, eds.), Gliwice: Silesian Technical University Press, 2001, (in Polish).

[18] L. Rutkowski, *Flexible Neuro-Fuzzy Systems. Structures, Learning and Performance Evaluation*, Boston: Kluwer Academic Publishers, 2004.

[19] M. Basseville and I.V. Nikiforov, *Detection of Abrupt Changes. Theory and Application*, London: Prentice Hall, 1993.

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

[21] E. Walter and L. Pronzato, *Identification of Parametric Models from Experimental Data*, Berlin: Springer-Verlag, 1997.

[22] J. Korbicz, J.M. Kościelny, Z. Kowalczuk, and W. Cholewa (eds.), *Process Diagnosis. Models, Methods of Artificial Intelligence, Applications*, Warsaw: WNT, 2002, (in Polish).

[23] R.J. Patton and J. Chen, "Observer-based fault detection and isolation: Robustness and applications", *Control Eng. Practice* 5, 671–682 (1997).

[24] B.D.O. Anderson and J.B. Moore, *Optimal Filtering*, New Jersey: Prentice Hall, 1979.

[25] J. Korbicz and P. Bidyuk, *State and Parameter Estimation. Digital and Optimal Filtering. Applications*, Zielona Góra: Technical University Press, 1993.

[26] Z. Kowalczuk and P. Suchomski, "Optimal detection observers based on eigenstructure assignment", in: *Fault Diagnosis. Models, Artificial Intelligence, Applications* (Korbicz J., Kościelny J.M., Kowlaczuk Z., Cholewa W., eds.), Berlin: Springer-Verlag, 219–260, (2004).

[27] J. Korbicz, Z. Fathi, and W.F. Ramirez, "State estimation schemes for fault detection and diagnosis in dynamic systems", *Int. J. Systems Sci.* 24 (5),985–1000 (1993).

[28] M. Boutayeb and D. Aubry, " A strong tracking extended Kalman observer for non-linear discrete time systems", *IEEE Trans. Automat. Control* 44, 1550–1556 (1999).

[29] A. Edelmayer, J. Bokor, Z. Szabo, and F. Szigeti, "Input reconstruction by means of system inversion. A geometric approach to fault detection and isolation in nonlinear systems", *Int. J. Appl. Math. and Comp. Sci.* 14 (2), 189–199 (2004).

[30] J.R. Koza, *Genetic Programming. On the Programming of Computers by Means of Natural Selection*, Cambridge: MIT Press, 1992.

[31] M.F. Metenidis, M. Witczak, and J. Korbicz, "A novel genetic programming approach to nonlinear system modelling. Application to the DAMADICS benchmark problem", *Eng. Appl. of Artificial Intelligence* 17, 363–370 (2004).

[32] R. Isermann (ed.), "Supervision, fault detection and diagnosis of technical systems", *Control Eng. Practice, Special Section* 5(5), (1997).

[33] D. Uciński, *Optimal Measurements Methods for Distributed Parameter System Identification*, New York: CRC Press, 2005.

[34] P.M. Frank and T. Marcu, "Diagnosis strategies and systems. Principles, fuzzy and neural approaches", in: *Intelligent Systems and Interfaces* (H.-N. Teodorescu, D. Mlynek, A. Kandel, and H.-J. Zimmermann, eds.), Boston: Kulwer, 2000.

[35] A. Obuchowicz, *Evolutionary Algorithms for Global Optimization and Dynamic System Diagnosis*, Zielona Góra: Lubuskie Scientific Society, 2003.

[36] M. Brown and C.J. Harris, *Neuro-Fuzzy Adaptive Modelling and Control*, London: Prentice Hall, 1994.

[37] D. Rutkowska, *Neuro-fuzzy Architectures and Hybrid Learning*, Heidelberg: Physica-Verlag, 2000.

[38] R.J. Patton, J. Korbicz, M. Witczak and F. Uppal, "Combined computational intelligence and analytical methods in fault diagnosis", in: *Intelligent Control Systems*, A.E. Ruano (ed.), London: IEE Press, pp. 349–392, 2005.

[39] W. Duch, J. Korbicz, L. Rutkowski, and R. Tadeusiewicz (eds.), *Biocybernetics and Biomedical Engineering 2000. Neural Networks*, Vol. 6, Akademicka Oficyna Wydawnicza, EXIT, Warsaw 2000, (in Polish).

[40] M.M. Gupta, L. Jin, and N. Homma, *Static and Dynamic Neural Networks*, New Jersey: John Wiley & Sons, 2003.

[41] S. Haykin, *Neural Networks. A Comprehensive Foundation*, Upper Saddle River: Prientice-Hall, 1999.

[42] S.J. Farlow (ed.), *Self-Organizing Methods in Modeling – GMDH Type Algorithms*, New York: Marcel Dekker, 1984.

[43] A.G. Ivakhnenko, "Polynominal theory of complex systems", *IEEE Trans. System, Man and Cybernetics* 1 (4), 44–58 (1971).

[44] D.T. Pham and L.Xing, *Neural Networks for Identification, Prediction and Control*, London: Springer-Verlag, 1995.

[45] M. Mrugalski, *Neural Network Based Modelling of Non-linear Systems in Fault Detection Schemes*, University of Zielona Góra, Faculty of Electrical Engineering, Computer Science and Telecommunications, doctoral dissertation, 2004, (in Polish).

[46] J.E. Mueller and F. Lemke, *Self-organising Data Maining*, Hamburg: Libri, 2000.

[47] L. Ljung, *System Identifivation. Theory for the Users*, New Jersey: Prentice-Hall, 1987.

[48] G.A.F. Seber, *Nonlinear Regression*, New York: John Wiley & Sons, 1989.

[49] S.H. Mo and J.P. Norton, "Fast and robust algorithm to compute exact polytope parameter bounds", *Math. and Comp. in Simulation* 32, 481–493, (1990).

[50] M. Witczak, J. Korbicz, M. Mrugalski and R.J. Patton, "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 (2006).

[51] J. Korbicz, M.F. Metendis, M. Mrugalski, and M. Witczak, "Confidence estimation of GMDH neural networks", in: *Artificial Intelligence and Soft Computing, ICAISC, Lecture Notes in Artificial Intelligence*, L. Rutkowski, J. Siekman, R. Tadeusiewicz, L.A. Zadeh (eds.), 3070, 210–216, (2004).

[52] J.M.F. Calado and J.M.G. Sá da Costa, "An expert system coupled with a hierarchical structure of fuzzy neural networks for fault diagnosis", *Appl. Math. and Comput. Science* 9 (3), 667–688, (1999).

[53] S-J.,C-H. Woo, H-S. Hwang, and K.B. Woo, "Evolutionary design of fuzzy rule base for nonlinear system modeling and control", *IEEE Trans. of Fuzzy Systems* 8 (1), 37–45 (2000).

[54] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programmes*, Berlin: Springer-Verlag, 1996.

[55] S. Chiu, "Fuzzy model identification based on cluster estimation", *Journal of Intelligent and Fuzzy Systems* 2 (3), 665–685 (1994).

[56] O. Nelles, *Nonlinear System Identification*, Berlin: Springer-Verlag, 2001.

[57] M. Kowal, *Optimization of Neuro-Fuzzy Structures in Technical Diagnostics Systems*, University of Zielona Góra, Faculty of Electrical Engineering, Computer Science and Telecommunications, doctoral dissertation, 2004, (in Polish).

[58] M. Kowal, J. Korbicz, M.J.G.C. Mendes, and J.M.F. Calado, "Fault detection using neuro-fuzzy networks", *Systems Science* 28 (1), 45–57 (2002).

[59] R. Babuska, *Fuzzy Modelling in Control*, London: Kluwer Academic Publishers, 1998.

[60] A. Piegat, *Fuzzy Modelling and Control*, Berlin: Springer-Verlag, 2001.

[61] M. Kowal and J. Korbicz, "Robust fault detection using neuro-fuzzy networks", *Proc. 16th IFAC World Congress*, Prague, Czech Republic (2005) (CD ROM).

[62] L. Rutkowski, *New Soft Computing Techniques for System Modelling, Pattern Classification and Image Processing*, Berlin: Springer-Verlag, 2004.

[63] DAMADICS, Website of the Reseach Training Network on the *Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems*, `http://diag.mchtr.pw.edu.pl/damadics`

[64] J.M. Kościelny and M. Bartyś, "Application of information system theory for actuator diagnosis", *Proc. 4th IFAC Symp. Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS, Budapest* 2, 949–954 (2000).

[65] J.C. Yang and D.W. Clarke, "The self-validating actuator", *Contr. Eng. Practice* 7, 249–260 (1999).

[66] M. Mrugalski, J. Korbicz, and R.J. Patton, "Robust fault detection via GMDH neural networks", *Proc. IFAC World Congress*, Prague, Czech Republic, (CD ROM).

[67] Z. Bubnicki, *Analysis and Decision Making in Uncertain Systems*, London: Springer-Verlag, 2004.