ARTIFICIAL INTELLIGENCE IN TECHNICAL DIAGNOSTICS*

Józef KORBICZ
Instytut Sterowania i Systemów Informatycznych, Uniwersytet Zielonogórski
ul. Podgórna 50, 65-246 Zielona Góra, tel. 068 328 24 22, fax: 068 328 47 51
e-mail: j.korbicz@issi.uz.zgora.pl

Summary
The paper deals with the problems of robust fault detection using soft computing techniques, particularly neural networks (Group Method of Data Handling, GMDH), neuro-fuzzy networks (Takagi-Sugeno (T-S) model) and genetic programming. The model-based approach to Fault Detection and Isolation (FDI) is considered. The main objective is to show how to employ the bounded-error approach to determine the uncertainty defined as a confidence range for the model output, the adaptive thresholds can be defined. Finally, the presented approaches are tested on a servoactuator being an FDI benchmark in the DAMADICS project.

Keywords: fault detection, robustness, adaptive threshold, neural networks, neuro-fuzzy networks, genetic programming.

SZTUCZNA INTELIGENCJAW DIAGNOSTYCE TECHNICZNEJ

Streszczenie
W artykule rozpatruje się problemy odpornej detekcji uszkodzeń z wykorzystaniem technik obliczeń inteligentnych, a w szczególności sieci neuronowych (Group Method of Data Handling, GMDH), sieci neuronowo-rozmytych (model Takagi-Sugeno) oraz programowania genetycznego. Rozpatruje się układ detekcji i lokalizacji uszkodzeń z modelem. Głównym celem jest pokazanie jak zastosować metodę ograniczonego błędu do wyznaczenia niepewności modeli neuronowych i rozmytych. Pokazano, że korzystając z wyznaczonych niepewnych modeli obliczeń inteligentnych zdefiniowanych w postaci przedziałów ufności dla wyjścia modelu można zdefiniować adaptacyjny próg decyzyjny. W ostatniej części efektywność rozpatrywanych podejść ilustrowana jest na przykładzie układu diagnostyki inteligentnego urządzenia siłownik-ustawnik-zawór z projektu DAMADICS.

Słowa kluczowe: detekcja uszkodzeń, odporność, próg adaptacyjny, sieci neuronowe, sieci neuronowo-rozmyte, programowanie genetyczne.

1. INTRODUCTION

The reliability, safe and availability of engineering systems and machines play an important role during their operation use. It is important especially nowadays, when industrial installations and control algorithms are becoming more and more sophisticated, and economics pressure to reduce the costs. An early detection of faults can help avoid the system shutdown or breakdown. A system that includes the capacity of detecting, isolating, identificating or classifying faults is called a fault detection and isolation system. In automatic control systems, defects may occur in sensors, actuators and/or components of the controlled process. Such faults appeared in a component may develop into failures of the whole systems and this effect can be easily amplified by the closed loop. Therefore, fault tolerant control systems has gained more and more importance in the last decade [1, 23]. The toleranece to faults can be achived by different strategies (see the excellent overview paper [51]), 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 FDI and a great variety of methods have been proposed. The core of modern diagnosis systems is the so-called model-based approach [11, 18, 19, 27, 40], in which

* This work has been supported by the Ministry of Science and Higher Education of the Republic of Poland under the project No. 1219/B/T02/2007/33.

either analytical and/or soft computing models are applied [15, 44]. Unfortunately, the analytical model-based approach is usually restricted to simpler systems described by linear models. When there are no mathematical models of the diagnosed system or the complexity of the dynamic model increases and the task of modeling is hard, an analytical model cannot be applied in the fault diagnosis system nor give satisfactory results. Therefore many efforts are made to use knowledge-based or qualitative or data-based models [3, 16, 41, 43]. They represent system behaviour in terms of heuristic or qualitative knowledge [22, 32]. The relationship between inputs and outputs may be described by a rule base or a set of parameters that have to be determined during an identification stage based on the learning data set. In this case databased models, such as neural networks [15, 24], fuzzy sets [20, 28, 31], the evolutionary algorithms [37, 50] or their combination [3, 43], can be considered.

Irrespective of the modelling method used (analytical or soft computing), there is always the problem of model uncertainty, i.e., the model-reality mismatch. Thus, the better model used to represent system behaviour, the better chance of improving the reliability and performance in diagnosing faults. Indeed, disturbances as well as model uncertainty are inevitable in industrial systems [42], and hence there exists a pressure creating the need for robustness in fault diagnosis systems [5]. This robustness requirement is usually achieved at the fault detection stage, i.e., the problem is to develop residual generators which should be insensitive (as far as possible) to model uncertainty and real disturbances acting on a system while remaining sensitive to faults. The most common approach to robust fault diagnosis is to use robust observers [5, 7, 10].

The main objective of this survey paper is to present recent developments in modern fault diagnosis with neural networks, neuro-fuzzy networks and genetic programming. In particular, the paper is organised as follows. Section 2 outlines the problem of model-based fault diagnosis. The problem of genetic programming in modelling in Section 3 is considered. Section 4 presents neural network-based approaches that can be used to settle the fault diagnosis problem when the mathematical state-space model is not available. The Neuro-Fuzzy (NF) structure optimization problem in Section 5 is considered. The so-called Bounded Error Approach (BEA) is applied for description of soft computing models uncertainty, i.e. for GMDH neural network and T-S fuzzy model. The final section presents a comprehensive study regarding the application of the approaches considered to the DAMADICS benchmark problem.

2. MODEL-BASED FAULT DIAGNOSIS

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, which consists of residual generation and evaluation.

In other words, three phases are distinguished in the process diagnosis [11, 18, 40] – 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.

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.

The analytical redundancy of measurement line exists when an additional value of process variable is obtained using mathematical model that connects the calculated variables with other measured signals. Analytical redundancy is used for fault detection where the analytical model of diagnosed systemismost important part.

Among known mathematical models a special role belongs to state space equations that are applied for designing of state observers [17]. A common disadvantage of analytical approaches to the FDI system is the fact that a precise mathematical model of the diagnosed process is required. An alternative solution can be obtained using soft computing techniques, i.e., artificial neural networks, fuzzy logic, expert systems and evolutionary algorithms [3, 48] or their combination as neuro-fuzzy networks [20, 43]. To apply soft computing modelling, empirical data, principles and rules which describe the diagnosed process are required [43].

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

3. GENETIC PROGRAMMING INMODELLING

Although there are many techniques for constructing non-analytical models, in one way or another, they finally boil down to several global optimization problems, like searching for an optimal model structure, the allocation of model parameters etc. They are nonlinear, multi-modal, usually multi-objective, so that conventional "local" optimization methods are insufficient to solve them. In recent years, direct search techniques, which are problem-independent, have been widely used in optimization. Unlike calculus-based methods (gradient descent, etc.), direct search algorithms do not require the use of derivatives. Gradient-descent methods work well when the objective surface is relatively smooth, with few local minima. However, realworld data are often multimodal and contaminated by noise which can further distort the objective surface.

Evolutionary Algorithms (EAs) are a broad class of stochastic optimization algorithms inspired by some biological processes which allow populations of organisms to adapt to their surrounding environment [33, 37]. Genetic Programming (GP) [30] is an extension of EAs.

3.1. Input-output representation of the system via GP

The set of possible candidate models from which the system model will be obtained constitutes an important preliminary task in any system identification procedure [13, 46].

Knowing that the diagnosed system exhibits nonlinear characteristics, a choice of the nonlinear model set must be made. In this section, an NARX (*Nonlinear AutoRegresive with eXogenous variable*) model was selected as the foundation for identification methodology. Let a Multi-Input and Multi-Output (MIMO) NARX model has the following form:

$$\begin{aligned}\hat{y}_{i,k} = g_i(&\hat{y}_{1,k-1},\ldots,\hat{y}_{1,k-n_{1,y}},\ldots,\hat{y}_{m,k-1},\ldots,\\ &\hat{y}_{m,k-n_{m,y}},\ldots,u_{1,k-1},\ldots,u_{1,k-n_1},\ldots,\\ &u_{r,k-1},\ldots,u_{r,k-n_{r,u}},\theta_i),\\ &i=1,\ldots m.\end{aligned} \tag{1}$$

Thus the system output is given by

$$y_k = \hat{y}_k + \varepsilon_k \tag{2}$$

where ε_k consists of a structural deterministic error, caused by the model-reality mismatch, and the measurement noise ν_k. The problem is to determine an unknown function $g(\cdot) = (g_1, \ldots, g_m)$ and to estimate the corresponding parameters vector $\theta = (\theta_1, \ldots, \theta_m)$.

One possible solution to this problem is the genetic programming approach. A tree is the main ingredient underlying the GP algorithm. In order to adapt GP to system identification it is necessary to represent the model (1) as a tree, or a set of trees [30]. The language of the trees in GP is formed by the user-defined function F set and the terminal T set, which form the nodes of the trees.

The functions should be chosen so that they are *a priori* useful in solving the problem, i.e., any knowledge concerning the system under consideration should be included in the function set. This function set is very important and should be universal enough to be capable of representing a wide range of nonlinear systems. The terminals are usually variables or constants. Moreover, a tree representation can be extended by the so-called *node gains*. A node gain is a numerical parameter associated with the node, which multiplies its output value.

One of the best known criteria which can be employed to select the model structure and to estimate its parameters is the Akaike Information Criterion (AIC) [46], where the following quality index is minimized:

$$J_{AIC}(M_i) = \frac{1}{2} J\left(M_i\left(\hat{\theta}^i\right)\right) + \frac{1}{n_T}\dim\theta^i, \tag{3}$$

where:

$$J\left(M_i\left(\theta^i\right)\right) = \ln\det\sum_{k=1}^{n_T}\varepsilon_k\varepsilon_k^T, \tag{4}$$

and $\hat{\theta}^i = \arg\min_{\theta^i} J\left(M_i\left(\theta^i\right)\right)$ are the obtained using the identification data set of n_T pairs of input/output measurements. The GP algorithm was successfully applied to identify the input-output model of the evaporation station at the Lublin Sugar Factor S. A. (Poland) [6]. Figure 1 illustrates the obtained results [50].

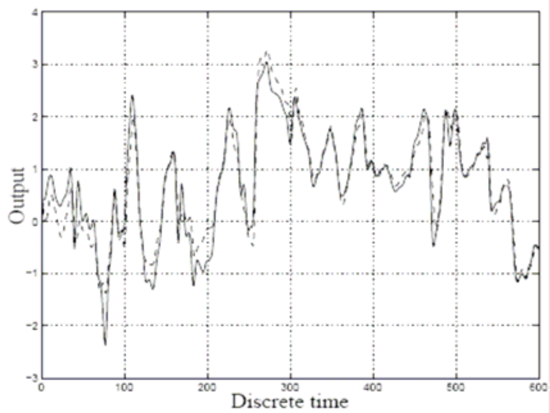

Fig. 1. System (solid line) and model (dashed line) output for the identification (left) and validation (right) data sets.

4. NEURAL NETWORKS IN FAULT DETECTION

For the model-based approach [18, 40], the neural network replaces the analytical model that describes the process under the normal operating conditions [15, 43]. First, the network has to be trained to settle this task. Learning data can be collected directly from the process, if possible, or from a simulation model that should be as realistic as possible. The latter possibility is of special interest for data acquisition in different faulty situations. This is especially important for the task of testing the residual generator because such data are not generally available from the real process. The training process can be carried out off-line or on-line (it depends on the availability of data) [9, 38].

The possibility to train a network on-line is very attractive, especially in the case of adapting a neural model to mutable environment or time-varying systems. After finishing the training, a neural network is ready for on-line residual generation. In order to be able to capture the dynamic behaviour of the system, the neural network should have dynamic properties [8, 39], e.g., it should be a recurrent network.

Residual evaluation is a decision-making process that transforms quantitative knowledge into qualitative *Yes* or *No* statements. It can also be perceived as a classification problem. The task is to match each pattern of the symptom vector with one of the pre-assigned classes of faults and the fault-free case. This process may be highly facilitated with intelligent decision making. To perform residual evaluation, neural networks can be applied, e.g., feed-forward networks or selforganizing maps [9, 14, 25].

4.1. Robust GMDH neural networks

A successful application of the ANNs to the system identification and fault diagnosis tasks [24] depends on a proper selection of the neural network architecture. In the case of the classical ANNs such as Multi-Layer Perceptron (MLP), the problem reduces to the selection of the number of layers and the number of neurons in a particular layer. If the obtained network does not satisfy prespecified requirements, then a new network structure is selected and parameter estimation is repeated once again. The determination of the appropriate structure and parameters of the model in the presented way is a complex task. Furthermore, an arbitrary selection of the ANN structure can be a source of model uncertainty. Thus, it seems desirable to have a tool which can be employed for automatic selection of the ANN structure, based only on the measured data. To overcome this problem, GMDH neural networks [12, 36] have been proposed. The synthesis process of the GMDH model 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 as to obtain the best quality approximation of the identified system. Thus, the task of designing a neural network is defined in such a way so as to obtain a model with a small uncertainty.

The idea of the GMDH approach relies on replacing the complex neural model by the set of hierarchically connected neurons. The behaviour of each neuron should reflect the behaviour of the system being considered. It follows from the rule of the GMDH algorithm that the parameters of each neuron are estimated in such a way that their output signals are the best approximation of the real system output. In this situation, the neuron should have the ability to represent the dynamics. One way out of this problem is to use dynamic neurons [8, 21, 39]. Dynamics in these neurons are realised by introducing a linear dynamic system – an IIR filter. The process of GMDH network synthesis leads to the evolution of the resulting model structure in such a way as to obtain the best quality approximation of the real system [49].

To obtain the final structure of the network, all unnecessary neurons are removed, leaving only those which are relevant to the computation of the model output. The procedure of removing the unnecessary neurons is the last stage of the synthesis of the GMDH neural network.

4.2. Confidence estimation of GMDH neural networks

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 [35] that the application of the classical evaluation criteria such as the Akaike Information Criterion (AIC) and the Final Prediction Error (FPE) [12] 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 leastsquare method to parameter estimation of neurons, a set of restrictive assumptions has to be satisfied [35]. An effective remedy to such a challenging problem is to use the bounded error approach [34]. Let us consider the following system:

$$y_k = r_k^T \theta + \varepsilon_k \tag{5}$$

where r_k stands the regressor vector, $\theta \in R^{n_\theta}$ denotes the parameter vector, and εk represents the difference between the original system and the model.

The problem is to obtain the parameter estimate vector $\hat{\theta}$, as well as the associated parameter uncertainty required to design robust fault detection system. The knowledge regarding the set of admissible parameter values allows obtaining the confidence region of the model output which satisfies

$$\tilde{y}_k^m \le y_k \le \tilde{y}_k^M \tag{6}$$

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

It is assumed that ε_k consists of a structural deterministic error caused by the model-reality mismatch, and the stochastic error caused by the measurement noise is bounded as follows:

$$\varepsilon_k^m \le \varepsilon_k \le \varepsilon_k^M \quad (7)$$

where the bounds ε_k^m and $\varepsilon_k^M\left(\varepsilon_k^m \ne \varepsilon_k^M\right)$ can be estimated [49].

The idea underlying the bounded-error approach is to obtain a feasible parameter set P [34] that is consistent with the input-output measurements used for parameter estimation. The resulting P is described by a polytope defined by a set of vertices V. Thus, the problem of determining the model output uncertainty can be solved as follows:

$$r_k^T \theta_k^m \le r_k^T \theta \le r_k^T \theta_k^M \quad (8)$$

were

$$\theta_k^m = \arg\min_{\theta \in V} r_k^T \theta, \qquad \theta_k^M = \arg\max_{\theta \in V} r_k^T \theta \quad (9)$$

As has been mentioned, the neurons in the l-th ($l > 1$) layer are fed with the outputs of the neurons from the ($l - 1$)-th layer. In order to modify the abovepresented approach for the uncertain regressor case, let us denote an unknown "true" value of the regressor $r_{n,k}$ by a difference between the measured value of the regressor r_k and the error in the regressor ε_k:

$$r_{n,k} = r_k - e_k \quad (10)$$

where it is assumed that the error e_k is bounded as:

$$e_{i,k}^m \le e_{i,k} \le e_{i,k}^M, \quad i = 1, \ldots, n_p \quad (11)$$

Using (5) and substituting (10) into (11), one can define the space containing the parameter estimates:

$$\varepsilon_k^m - e_k^T \theta \le y_k - r_k^T \theta \le \varepsilon_k^M - e_k^T \theta \quad (12)$$

which makes it possible to adapt the above-described technique to the error-in-regressor case [49].

The proposed modification of the BEA makes it possible to estimate the parameter vectors of the neurons from the l-th, l > 1 layers. Finally, it can be shown that the model output uncertainty has the following form:

$$\tilde{y}_k^m \le r_n^T \theta \le \tilde{y}_k^M \quad (13)$$

In order to adapt the presented approach to parameter estimation of non-linear neurons with an activation function $\xi(\cdot)$, it is necessary to transform the relation:

$$\varepsilon_k^m \le y_k - \xi r_k^T \theta \le \varepsilon_k^M \quad (14)$$

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

$$\xi^{-1}\left(y_k - \varepsilon_k^M\right) \le r_k^T \theta \le \xi^{-1}\left(y_k - \varepsilon_k^m\right). \quad (15)$$

Knowing the model structure and possessing the knowledge regarding its uncertainty, it is possible to design a robust fault detection scheme with an adaptive threshold (Fig. 2).

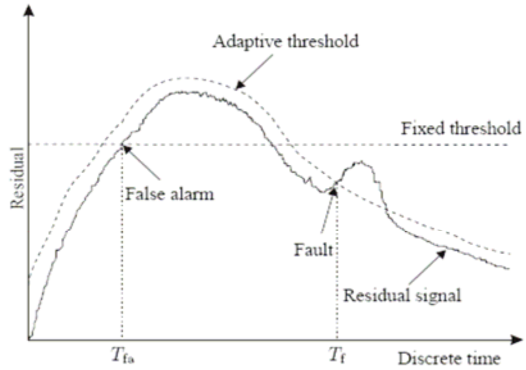

Fig. 2. Illustration of the concept of the adaptive threshold

The model output uncertainty interval, calculated with the application of the GMDH model, should contain the real system response in the fault-free mode. Therefore, the system output should satisfy:

$$\tilde{y}_k^m + \varepsilon_k^m \le y_k \le \tilde{y}_k^M + \varepsilon_k^M \quad (16)$$

This means that robust fault detection boils down to checking if the output of the systemsatisfies (16). Thus, when (16) is violated, then a fault symptom occurs.

5. NEURO-FUZZY NETWORKS IN FAULT DETECTION

The procedure of neuro-fuzzy network design consists of the structure selection stage and the parameter estimation stage [26, 44, 45]. The pessimistic scenario assumes the construction of the neuro-fuzzy network only on the basis of the available measurements. The main problem is to obtain the required accuracy and transparency of the rule base in such a situation. A lot of different methods have already been developed both for structure selection and parameter estimation of the neuro-fuzzy network, but there is a demand for better, more effective algorithms, and active research is still conducted in this area.

Takagi-Sugeno neuro-fuzzy networks can be viewed as multi-model systems which consist of some rules, and each rule defines a single model as the consequent of the rule [28, 29, 44]. The global neuro-fuzzy system is a set of N_r partial models, where N_r determines the number of fuzzy rules. The output of the global system is calculated as a mixture of partial model outputs. The rule fulfillment is determined by fuzzy sets. In order to ensure the desired accuracy of the neurofuzzy system, the membership functions of fuzzy sets must be placed properly in the input space, the number of rules must be appropriate and the parameters of partial models must be chosen to minimize the defined error.

5.1. Robust neuro-fuzzy networks

The application of neuro-fuzzy networks in diagnostic areas [3, 4] creates a demand for suitable design procedures which would take into account the specificity of the fault diagnosis task. An important problem from the diagnostic point of view is residual confidence interval minimization because it makes it possible to detect a fault appropriately early. It has to be stressed that the value of the confidence interval for residuals depends directly on the uncertainty of the model which is used to generate the residuals. If the confidence interval is not consistent with model uncertainty, the fault detection system can trigger off a lot of false alarms. It is obvious in such a situation that model uncertainty has to be considered in fault detection threshold calculations [5, 47, 48]. It is also important to minimize model uncertainty in order to obtain a reliable fault detection system that would be able to detect a fault fast and at an early stage. So special procedures for neuro-fuzzy model design must be developed.

To overcome the problem, an alternative approach in the form of the BEA method [34] can be applied to tune the parameters of the Takagi-Sugeno neuro-fuzzy network and to calculate the admissible set of parameters and the confidence interval for the network output. The method requires only the information about the range of the disturbances which corrupt measurements. The application of the BEA algorithm for computing the confidence interval of the Takagi-Sugeno fuzzy model output requires to establish some assumptions in order to view the model in the form of an Linear in Parameter (LP) system [28, 29]. The main assumption based on the fact that the parameters of the membership functions of the fuzzy sets are known. Appropriate selection of the values of these parameters has an essential influence on the uncertainty of the whole fuzzy model. Wrong values of these parameters can significantly increase model uncertainty, thus the model can be unsuitable for diagnostic tasks. In order to present the BEA approach for estimating the parameters of the determining dynamic T-S network, let us consider the following model:

$$y_k = \sum_{i=1}^{n} \phi_{i,k}, y_{i,k} \tag{17}$$

where $y_{i,k}$ is the output od the i-th rule and

$$\phi_{i,k} = \frac{\mu_{i,k}}{\sum_{j=1}^{n} \mu_{j,k}} . \tag{18}$$

The model described by the equation (17) can be viewed in the form of an LP system:

$$y = x_k^T \theta \tag{19}$$

where

$$x_k = \begin{bmatrix} \phi_{1,k} z_1^k \\ \phi_{2,k} z_2^k \\ \vdots \\ \phi_{n,k} z_n^k \end{bmatrix}, \quad \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix},$$

if the parameters of the fuzzy sets are treated like constant values. Here z_i^k denotes the input vector containing the delayed inputs u_i^k of the local models and the delayed output y_i^k of the local models, i.e., $z_i^k = \left[y_i^k, y_i^{k-1}, \ldots, y_i^{k-n_a}, y_i^{k-1}, y_i^{k-2}, y_i^{k-n_b}, \right]$. The output error is given by the following formulae:

$$\varepsilon_k^m \le \varepsilon_k \le \varepsilon_k^M . \tag{21}$$

thus the admissible set of parameters for N data points is given by the following expression:

$$P = \left\{ \theta \in R^n \middle| y_k^{'} - \varepsilon^M \le x_k^T \theta \le y_k^{'} - \varepsilon^m, k = 1, \ldots N \right\} \tag{22}$$

Each point inside the set P defines the vector of model parameters and all sets of parameters determine the group of models consistent with the measurements and bounds. This means that, instead of one model, a set of models with different parameters is given and the output signal is represented in the form of an interval which contains all possible model responses. Real applications usually require a single output value, thus one set of parameters must be chosen. The most common approach chooses the geometrical center of the area P as the set of parameters that is used to calculate the output of the model. This sample procedure is shown in Fig. 3. If the maximum and minimum values of the parameters are known:

$$\theta_i^m = \arg\min_{\theta \in P} \theta_i , \tag{23}$$

$$\theta_i^M = \arg\max_{\theta \in P} \theta_i , \tag{24}$$

the estimates of the parameters can be computed using the following formula:

$$\theta_i = \frac{\theta_i^m + \theta_i^M}{2}, \quad i = 1, \ldots, N \tag{25}$$

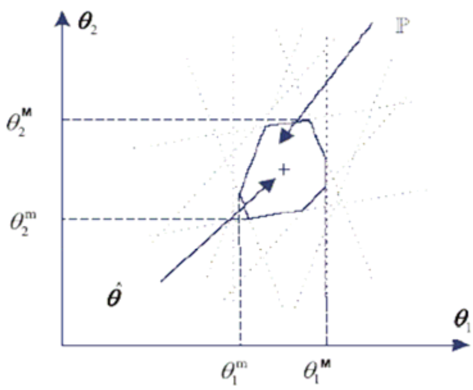

Fig. 3. Sample set of parameters P

The minimum and maximum values for the following parameters are determined using the linear programming technique [34]. The feasible set of parameters is used also to compute the confidence interval for the output of the system:

$$x_k^T \theta_k^m + \varepsilon^m \le y_k^{'} \le x_k^T \theta_k^M + \varepsilon^M , \tag{26}$$

were

$$\theta_k^M = \arg\max_{\theta \in W} x_k^T \theta \tag{27}$$

$$\theta_k^m = \arg\min_{\theta \in W} x_k^T \theta \tag{28}$$

The confidence interval can be directly applied to calculate the adaptive threshold for the residual signal:

$$r_k = y_k^{'} - y_k \,. \tag{29}$$

Finally, the adaptive threshold is described by the following inequalities:

$$x_k^T \theta_k^m + \varepsilon^m - y_k \le e_{r,k} \le x_k^T \theta_k^M + \varepsilon_k^M - y_k \tag{30}$$

Unfortunately, the computations required to determine all verticesWof the convex polyhedron *P* are so time and memory consuming that it is hard to employ the classical BEA algorithm for complicated models. In this case the methods that approximate the actual set *P* by the area which has a simplified shape should be employed [34]. One of the proposed solution is the Outer Bounding Ellipsoid (OBE) method which has been applied to fault detection in a DC engine [29].

6. NEURO-FUZZY-BASED FAULT DETECTION OF AN INTELLIGENT ACTUATOR

The scheme of the actuator with an intelligent positioner is given in Fig. 4. Such actuator has been investigated by international research group during the realization of the so-called DAMADICS [6] project. In the Fig. 6 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 setpoint (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).

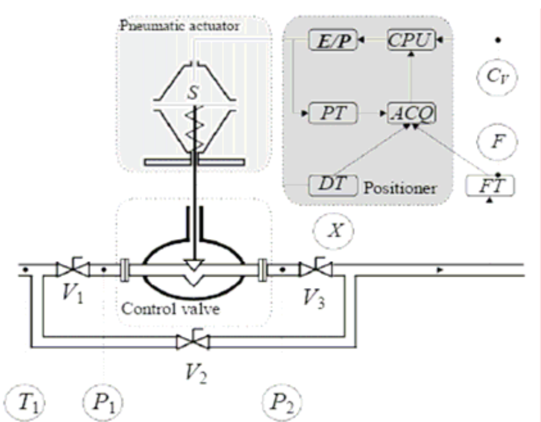

Fig. 4. Scheme of the intelligent actuator

Applying the method for structure generation of NF models [28, 29] and the results presented in Subsection 4.1, 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 5 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. 6, 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 [6] was tested as well.

Table 2. Fault detection results
(S-small, M-medium, B-big, I-incipient)

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_{12}	Pressure drop	Y	Y	Y	
f_{13}	Unexpected pressure change			Y	Y
f_{14}	Opened bypass valves	Y	Y	Y	Y
f_{15}	Flow rate sensor fault F	Y	Y	Y	

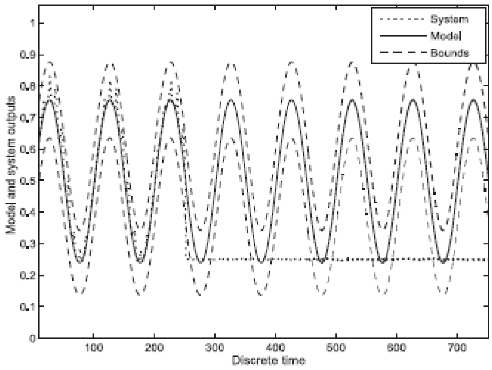

Fig. 5. Model and system output with system output bounds (big fault)

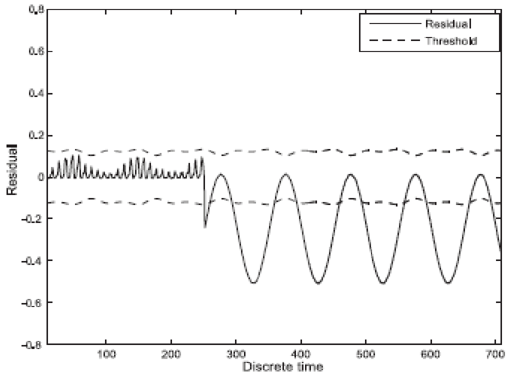

Fig. 6. Residual and adaptive threshold (big fault)

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.

7. CONCLUSIONS

From the view point of engineering, it is clear that providing fast and reliable fault detection and isolation is an integral part of control design, particularly as far as the control of complex industrial systems is considered. The main objective of this paper was to consider a robust model-based fault detection system applying soft computing models. Special attention was paid to the uncertainty of such models 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 T-S 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 neuro-fuzzy models.

LITERATURA

1. Blanke M., Kinnaert M., Lunze J. and Staroswiecki M. (2003): *Diagnosis and Fault-Tolerant Control.* New York: Springer-Verlag.
2. Boutayeb M. and Aubry D. (1999): *A strong tracking extended Kalman observer for nonlinear discrete-time systems*. IEEE Trans. Automatic Control, Vol. 44, No. 8, pp. 1550–1556.
3. Calado J. M. F, Korbicz J., Patan K., Patton R. J. and Sa da Costa J. M. G (2001): *Soft computing approaches to fault diagnosis for dynamic systems*. European Journal of Control, Vol. 7, No. 2-3, pp. 248–286.
4. Calado J. M. F., Sa da Costa J. M. G., Bartys M. and Korbicz J. (2006): *FDI approach to the DAMADICS benchmark problem based on qualitative reasoning coupled with fuzzy neural networks*. Control Engineering Practice, Vol. 14, No. 6, pp. 685–698.
5. Chen J. and Patton R. J. (1999): *Robust Model Based Fault Diagnosis for Dynamic Systems.* London: Kluwer Academic Publishers.
6. DAMADICS (2002): *Website of the RTN DAMADICS: Development and Application of Methods for Actutator Diagnosis in Industrial Control Systems* http://diag.mchtr.pw.edu.pl/damadics
7. Frank P. M., Schreier G. and Garcia E. A. (1999): *Nonlinear observers for fault detection and isolation*, In: New Directions in Nonlinear Observer Design (Nijmeijer H. and T. Fossen, Eds.) Berlin: Springer-Verlag.
8. Gupta M. M., Jin L. and Homma N. (2003): *Static and Dynamic Neural Networks. From Fundamentals to Advanced Theory.* New Jersey: Wiley.
9. Haykin S. (1999): *Neural Networks. A omprehensive Foundation*. New Jersey: Prentice-Hall.
10. Hui S. and ˙Zak S. H. (2005): *Observer design for systems with unknown input*. Int. J. Appl. Math. and Comput. Sci., Vol. 15, No. 4, pp. 431–446.
11. Isermann R. (2006): *Fault Diagnosis Systems. An Introduction from Fault Detection to Fault Tolerance*. New York: Springer-Verlag.
12. Ivakhnenko A. G. and Mueller J. A. (1995): *Selforganizing of nets of active neurons*. System Analysis Modelling Simulation, Vol. 20, pp. 93–106.

13. Janczak A. (2005): *Identification of Nonlinear Systems Using Neural Networks and Polynomial Models. A Block-oriented Approach*. Lecture Notes in Control and Information Sciences, Vol. 310, Berlin: Springer-Verlag.
14. Koivo H. M. (1994): *Artificial neural networks in fault diagnosis and control*. Control Engineering Practice, Vol. 2, No. 7, pp. 90-101.
15. Korbicz J. (2006): *Fault detection using analytical and soft computing methods*. Bulletin of the Polish Academy of Sciences: Technical Sciences, Vol. 54, No. 1, pp. 75–88.
16. Korbicz J. (Ed.) (2006): *Soft Computing in Control and Fault Diagnosis*. Special section in Int. J. Appl. Math. and Comput. Sci., Vol. 16, No. 1, pp. 7–114.
17. Korbicz J. and Bidyuk P. I. (1993): *State and Parameter Estimation. Digital and Optimal Filtering, Applications.* Technical University of Zielona Góra Press.
18. Korbicz J., Kościelny J. M., Kowalczuk Z. and Cholewa W. (Eds.) (2004): *Fault Diagnosis. Models, Artificial Intelligence, Applications*. Berlin: Springer-Verlag.
19. Korbicz J., Kościelny J. M., Kowalczuk Z. and Cholewa W. (Eds.) (2002): *Diagnostyka procesów. Modele, Metody sztucznej inteligencji, Zastosowania*. Warszawa: Wydawnictwo Naukowo Techniczne, WNT, (in Polish).
20. Korbicz J., Kowal M. (2007): *Neuro-fuzzy networks and their application to fault detection of dynamical systems*. Engng. Appl. of Artif. Intelligence, Vol. 20, pp. 609–617.
21. Korbicz J. and Mrugalski M. (2008): *Confidence estimation of GMDH neural networks and its application in fault detection systems*. Int. J. System Science, (accepted).
22. Korbicz J., Patan K. and Kowal M. (Eds.) (2007): *Diagnostics of Processes and Systems*.Warszawa: Akademicka Oficyna Wydawnicza EXIT, (in Polish).
23. Korbicz J., Patan K and Kowal M. (Eds.) (2007): *Fault Diagnosis and Fault Tolerant control*. Warszawa: Akademicka Oficyna Wydawnicza EXIT.
24. Korbicz J., Patan K. and Obuchowicz A. (2001): *Neural network fault detection system for dynamic processes*. Bulletin of the Polish Academy of Sciences. Technical Sciences., Vol. 49, No. 2, pp. 301–321.
25. Korbicz J., Obuchowicz A. and Uciński D. (1994): *Artificial Neural Networks. Fundations and Applications*. Warsaw: Akademicka Oficyna Wydawnicza EXIT, (in Polish).
26. Korbicz J. and Kowal M. (2007): *Neuro-fuzzy networks and their application to fault detection of dynamical systems*. Engng. Appl. Artif. Intelligence, Vol. 20, pp. 609-617.
27. Kościelny J. M. (2001): *Diagnostyka zautomatyzowanych procesów przemysłowych*. Warszawa: Akademicka Oficyna Wydawnicza EXIT, (in Polish).
28. Kowal M. (2005): *Optimization of Neuro-Fuzzy Structures in Technical Diagnostics*. Lecture Notes in Control and Computer Science, Vol. 9, University of Zielona Góra Press.
29. Kowal M. and Korbicz J. (2006): *Robust fault detection using neuro-fuzzy models.* Przegląd Elektrotechniczny, No. 1, pp. 32–36.
30. Koza J. R. (1992): *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* Cambridge: The MIT Press.
31. Köppen-Seliger B. and Frank P. M. (1999): *Fuzzy logic and neural networks in fault detection*. In: Fusion of Neural Networks, Fuzzy Sets, and Genetic Algorithms (L. Jain and N. Martin, Eds.). New York: CRC Press, pp. 169–209.
32. Ligeza A., Fuster-Parra P. (1997): *And/or/not causal graph – a model for diagnostic reasoning*. Appl. Math. and Comput. Sci., Vol. 7, No. 1, pp. 183–203.
33. Michalewicz Z. (1996): *Genetic Algorithms + Data Structures = Evolution Programs.* Berlin: Springer-Verlag.
34. Milanese M., Norton J., Piet-Lahanier H. and Walter E. (1996): *Bounding Approaches to System Identification.* New York: Plenum Press.
35. MrugalskiM. (2003): *Neural Network Based Modelling of Nonlinear Systems in Fault Detection Schemes.* Ph.D. thesis, University of Zielona Góra, Faculty of Electrical Engineering, Computer Science and Telecommunications, (in Polish).
36. Mrugalski M. and Korbicz J. (2005): *Robust fault detection via GMDH neural networks*. Proc. 16-th IFAC World Congress, Prague, Czech Republic, CD-ROM.
37. Obuchowicz A. (2003): *Evolutionary Algorithms for Global Optimization and Dynamic System Diagnosis.* Zielona Góra: Lubuskie Towarzystwo Naukowe.
38. Osowski S. (2006): *Artificial Neural Networks for Information Processing*. Warsaw: Oficyna Wydawnicza Politechniki Warszawskiej, (in Polish).
39. Patan K. (2007): *Stability analysis and the stabilization of a class of discrete-time dynamic neural networks*. IEEE Trans. Neural Networks, Vol. 18, No. 3, pp. 660-673.
40. Patton R. J., Frank P. M. and Clark R. N. (2000): *Issues of Fault Diagnosis for Dynamic Systems.* Berlin: Springer-Verlag.
41. Patton R. J. and Korbicz J. (Eds.) (1999): *Advances in Computational Intelligence for Fault Diagnosis Systems*. Int. J. Appl. Math. and Comput. Sci., Special Issue, Vol. 9, No. 3, pp. 471–735.
42. Patton R. J., Korbicz J. and Lesecq S. (Eds.) (2006): *A Benchmark Study of Fault Diagnosis for an Industrial Actuator*. Control Engineering

Practice, Special Issue, Vol. 14, No. 6, pp. 575–717.

43. Patton R. J., Korbicz J., Witczak M. and Uppal F. (2005): *Combined computational intelligence and analytical methods in fault diagnosis*. In: Intelligent Control Systems using Computational Intelligence Techniques (Rauno A. E. (Ed.). London: The IEE Press, pp. 349–392.
44. Piegat A. (2003): *Modelling and Fuzzy Systems.* Warszawa: Akademicka Oficyna Wydawnicza EXIT, (in Polish).
45. Rutkowska D. (2002): *Neuro-Fuzzy Architectures and Hybrid Learning.* Heidelberg, New York: Physica-Verlag, Springer-Verlag Company.
46. Walter E. and Pronzato L. (1997): *Identification of Parametric Models from Experimental Data*. London: Springer.
47. Witczak M. (2007): *Modelling and Estimation Strategies for Fault Diagnosis of Non-linear Systems*. Lecture Notes in Control and Information Sciences, No. 354, Berlin, Heidelberg: Springer.
48. Witczak M. (2006): *Advances in model-based fault diagnosis with evolutionary algorithms and neural networks*. Int. J. Appl. Math. and Comput. Sci., Vol. 16, No. 1, pp. 85–99.
49. Witczak M., Korbicz J., Mrugalski M. and Patton R. J. (2006): *A GMDH neural network-based approach to robust fault diagnosis: application to the DAMADICS benchmark problem*. Control Engineering Practice, Vol. 14, No. 6, pp. 671–683.
50. Witczak M., Obuchowicz A. and Korbicz J. (2002): *Genetic programming based approaches to identification and fault diagnosis of non-linear dynamic systems*. Int. J. Control, Vol. 75, No. 13, pp. 1012–1031.
51. Zhang Y. (2007): *Active fault-tolerant control systems: integration of fault diagnosis and reconfigurable control*, In: Fault Diagnosis and Fault Tolerant Control. (Korbicz J., Patan K. and Kowal M., Eds.), Warszawa: Akademicka Oficyna Wydawnicza EXIT, pp. 21–41.