Int. J. Appl. Math. Comput. Sci., 2006, Vol. 16, No. 1, 85–99

ADVANCES IN MODEL–BASED FAULT DIAGNOSIS WITH EVOLUTIONARY ALGORITHMS AND NEURAL NETWORKS

MARCIN WITCZAK

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

Challenging design problems arise regularly in modern fault diagnosis systems. Unfortunately, the classical analytical techniques often cannot provide acceptable solutions to such difficult tasks. This explains why soft computing techniques such as evolutionary algorithms and neural networks become more and more popular in industrial applications of fault diagnosis. The main objective of this paper is to present recent developments regarding the application of evolutionary algorithms and neural networks to fault diagnosis. In particular, a brief introduction to these computational intelligence paradigms is presented, and then a review of their fault detection and isolation applications is performed. Close attention is paid to techniques that integrate the classical and soft computing methods. A selected group of them is carefully described in the paper. The performance of the presented approaches is illustrated with the use of the DAMADICS fault detection benchmark that deals with a valve actuator.

Keywords: fault diagnosis, evolutionary algorithms, neural networks, robustness

1. Introduction

A continuous increase in the complexity, efficiency, and reliability of modern industrial systems necessities a continuous development of the control and fault diagnosis theory and practice (Blanke *et al.*, 2003; Korbicz *et al.*, 2004). These requirements extend beyond normally accepted safety-critical systems of nuclear reactors and chemical plants or aircrafts to new systems such as autonomous vehicles or fast rail systems. Early detection and maintenance of faults can help avoid the system shutdown, breakdown and even catastrophes involving human fatalities and material damage. A modern control system that is able to tackle such a challenging problem is presented in Fig. 1.

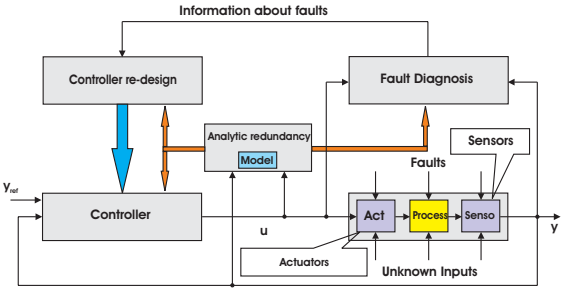


Fig. 1. Modern control system.

As can be observed, the controlled system is the main part of the scheme, and it is composed of actuators, process dynamics and sensors. Each of these parts is affected by the so-called unknown inputs that can be perceived as process and measurement noise as well as external disturbances acting on the system. When model-based control and diagnosis are utilized (Blanke *et al.*, 2003; Korbicz *et al.*, 2004), then the unknown input can also be extended by model uncertainty, i.e., the mismatch between a model and the system being considered. The system may also be affected by faults, which can be divided into three groups, i.e., actuator faults, component (or process) faults, and sensor faults. The role of the fault diagnosis part is to monitor the behaviour of the system and to provide all possible information regarding the abnormal functioning of its components. As a result, the overall task of fault diagnosis consists of three subtasks: fault detection, fault isolation, and fault identification (Chen and Patton, 1999). However, from a practical viewpoint, to pursue a complete fault diagnosis, the following three steps have to be realized (Frank and Köppen-Seliger, 1997):

Residual generation: generation of the signals that reflect the fault. Typically, the residual is defined as a difference between the system and model outputs.

Residual evaluation: logical decision making at the time of the occurrence and location of faults.

Fault identification: determination of the type of a fault, its size and cause.

Knowledge resulting from these steps is then provided to the controller re-design part, which is responsible for changing the control law in such a way as to maintain a required system performance. Thus, the scheme presented in Fig. 1 can be perceived as a fault-tolerant one.

Fault-Tolerant Control (FTC) (Blanke *et al.*, 2003) is one of the most important research directions underlying contemporary control engineering. FTC can also be perceived as an optimized integration of advanced fault diagnosis (Korbicz *et al.*, 2004; Witczak, 2003) and control (Blanke *et al.*, 2003) techniques. There is no doubt that the theory (and practice, as a consequence) of fault diagnosis and control is well-developed and mature for linear systems only (Chen and Patton, 1999; Korbicz *et al.*, 2004). There are also a number of different approaches that can be employed to settle the robustness problems regarding model uncertainty (Chen and Patton, 1999; Korbicz *et al.*, 2004). Such robustness is especially important in practical implementations where various sources of uncertainty may be present, e.g., differences between different copies of a given component, time-varying properties, noise, external disturbances, etc. Another kind of solutions that may increase the performance of the FTC scheme is based on appropriate scheduling of control test signals in such a way as to gain as much information as possible about the system being supervised (Delebecque *et al.*, 2003). Unfortunately, this technique can be used for linear systems only. In the light of the above discussion, it is clear that the development of new approaches for non-linear systems is justified. From this discussion, it is also clear that fault diagnosis has become an important issue in the theory and practice of modern automatic control. This justifies the objectives of this paper, which aims at presenting the latest developments in fault diagnosis with evolutionary algorithms and neural networks.

In spite of the fact that a large spectrum of analytical techniques for Fault Detection and Isolation (FDI) of non-linear systems can be found in the literature (Blanke *et al.*, 2003; Chen and Patton, 1999; Korbicz *et al.*, 2004), they usually all suffer from the lack of an appropriate mathematical description of the system being considered. If there are no sufficiently accurate analytical models, then the one feasible way is to use the so-called soft computing techniques (Korbicz *et al.*, 2004). A large amount of knowledge on using these techniques for model-based fault diagnosis has been accumulated through the literature since the beginning of the 1990s (see, e.g., (Chen and Patton, 1999; Frank and Köppen-Seliger, 1997; Korbicz *et al.*, 2004; Ruano, 2005; Witczak, 2003) and the references therein). One objective of this paper is to show the benefits that can be gained while using these techniques in practice. Another objective is to show that the integration of soft computing and classical techniques results in hybrid fault diagnosis techniques that are superior to the classical ones.

The paper is organised as follows: Section 2 is devoted to the application of evolutionary algorithms to fault diagnosis, while in Section 3 the problem of fault diagnosis is settled with neural networks. Both sections present examples of integrating soft computing and analytical techniques. The presented approaches are illustrated with results concerning the DAMADICS benchmark (DAMADICS, 2004). Finally, the closing section concludes the paper.

2. Evolutionary Algorithms in FDI

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. Such algorithms have been influenced by Darwin's theory of natural selection, or the survival of the fittest (published in 1859). The idea behind it is that only certain organisms can survive, i.e., only those which can adapt to the environment and win the competition for food and shelter. Almost at the same time that Darwin's theory was presented (1865), Mendel published a short monograph about experiments with plant hybridisation. He observed how traits of different parents are combined into offspring by sexual reproduction. Darwinian evolutionary theory and Mendel's investigations of heredity in plants became the foundations of evolutionary search methods and led to the creation of the neo-Darwinian paradigm (Fogel, 1995).

Evolutionary algorithms have been the subject of an uncountable number of papers (Bäck *et al.*, 1997). Generally, there is a large number of different kinds of EAs and the most popular of them are: genetic algorithms (GAs) (Holland, 1975), genetic programming (GP) (Gray *et al.*, 1998; Koza, 1992), evolutionary programming (Fogel *et al.*, 1999), evolutionary strategies (Michalewicz, 1996), and evolutionary search with soft selection (Galar, 1989).

Although the origins of evolutionary algorithms can be traced back to the late 1950s (see Bäck *et al.*, 1997 for a comprehensive introduction and survey on EAs), the first works on evolutionary algorithms in control engineering were published at the beginning of the 1990s. In 2002, Fleming and Purshouse (Fleming and Purshouse, 2002) tackled a challenging task of preparing a comprehensive survey on the application of evolutionary algorithms to control engineering. As is indicated in (Fleming and Purshouse, 2002), there are relatively scarce publications on applications of evolutionary algorithms to the design of FDI systems.

This paper, rather than providing an exhaustive survey on evolutionary algorithms in fault diagnosis, is aimed at providing a comprehensive account of the published work that exploits the special nature of EAs. This means that the works dealing with EAs applied as alternative optimisers, e.g., for training neural and/or fuzzy systems are not included here. In other words, the main objective is to extend the material of (Fleming and Purshouse, 2002) by introducing the latest advances in fault diagnosis with evolutionary algorithms.

Irrespective of the identification method selected for designing the model, there always exists the problem of model uncertainty, i.e., the model-reality mismatch. To overcome this problem, many approaches have been proposed (Chen and Patton, 1999; Korbicz *et al.*, 2004). Undoubtedly, the most common approach is to use robust observers, such as the Unknown Input Observer (UIO) (Chen and Patton, 1999; Korbicz *et al.*, 2004; Witczak, 2003), which can tolerate a degree of model uncertainty and hence increase the reliability of fault diagnosis. In such an approach, the model-reality mismatch can be represented by the so-called unknown input. Hence the state estimate and, consequently, the output estimate are obtained taking into account model uncertainty. Unfortunately, when the direction of faults is similar to that of an unknown input, then the unknown input decoupling procedure may considerably impair fault sensitivity. In order to settle this problem, Chen *et al.* (1996) (see also Chen and Patton, 1999) formulated observer-based FDI as a multiobjective optimisation problem, in which the task was to maximise the effect of faults on the residual, whilst minimising the effect of an unknown input. The approach was applied to the detection of sensor faults in a flight control system. A similar approach was proposed by Kowalczuk *et al.* (1999), where the observer design is founded on a Pareto-based approach, in which the ranking of an individual solution is based on the number of solutions by which it is dominated. These two solutions can be applied to linear systems only.

In spite of the fact that a large amount of knowledge on designing observers for non-linear systems has been accumulated through the literature since the beginning of the 1970s, a customary approach is to linearize the non-linear model around the current state estimate, and then to apply techniques for linear systems, as is the case for the extended Kalman filter (Korbicz *et al.*, 2004). Unfortunately, this strategy works well only when linearization does not cause a large mismatch between the linear model and non-linear behaviuor. To improve the effectiveness of state estimation, it is necessary to restrict the class of non-linear systems while designing observers. Unfortunately, the analytical design procedures resulting from such an approach are usually very complex, even for simple laboratory systems (Zolghardi *et al.*, 1996). To overcome this problem, Porter and Passino proposed the so-called genetic adaptive observer (Porter and Passino, 1995). They showed how to construct such an observer, where a genetic algorithm evolves the gain matrix of the observer in real time so that the output error is minimized. Apart from the relatively simple design procedure, the authors did not provide convergence conditions of the observer. They did not consider the robustness issues with respect to model uncertainty either. A solution that does not posses such drawbacks was proposed by Witczak *et al.* (2002). In particular, the authors showed the convergence condition of the observer, and proposed a technique for increasing its convergence rate with genetic programming. This approach will be detailed in Section 2.2. It should be strongly underlined that 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. This explains why most of the examples considered in the literature are devoted to simulated or laboratory systems, e.g., the celebrated two- (three- or even four-) tank system, an inverted pendulum, a travelling crane, etc. To tackle this problem, a genetic programming-based approach for designing state-space models from input-output data was developed in (Witczak *et al.*, 2002; Witczak, 2003). This approach will be detailed in Section 2.1. A further development of this technique related to input-output models was proposed in (Metenidis *et al.*, 2004).

Evolutionary algorithms have also been applied to FDI methods that are not based on the concept of residuals. Marcu (1997) formulated FDI design as a feature selection and classifier design problem. EA has also been applied to the generalised task of determining the fault from a collection of symptoms (Miller, 1993). The method relied upon the availability of *a-priori* probabilities that a particular fault caused a particular symptom. In (Chen *et al.*, 2003), the authors employed genetic algorithms-based evolutionary strategy for fault diagnosis-related classification problems, which includes two aspects: evolutionary selection of training samples and input features, and evolutionary construction of the neural network classifier. Finally, Sun *et al.* (2004) used the bootstrap technique to preprocess operational data acquired from a running diesel engine, and the genetic programming approach to find the best compound feature that can discriminate between the four kinds of commonly operating modes of the engine.

2.1. System Identification for FDI

Let us consider the following class of non-linear discrete-time systems:

$$\boldsymbol{x}_{k+1} = g\left(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{p}\right) + \boldsymbol{w}_k, \tag{1}$$

$$\boldsymbol{y}_{k+1} = \boldsymbol{C}_{k+1}\boldsymbol{x}_{k+1} + \boldsymbol{v}_k, \tag{2}$$

where $\boldsymbol{p} \in \mathbb{R}^{n_p}$ is the parameter vector, $\boldsymbol{x}_k \in \mathbb{R}^n$ is the state vector, $\boldsymbol{u}_k \in \mathbb{R}^r$ is the input vector, $\boldsymbol{y}_k \in \mathbb{R}^m$ is the output vector, $g(\cdot)$ is a non-linear function, $\boldsymbol{w}_k \in \mathbb{R}^n$ and $\boldsymbol{v}_k \in \mathbb{R}^m$ are the process and measurement noise, respectively. With a slight abuse of notation, the parameter vector will be neglected in model equations. Assume that the function $\boldsymbol{g}(\cdot)$ has the form

$$\boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \boldsymbol{A}(\boldsymbol{x}_k)\boldsymbol{x}_k + \boldsymbol{h}(\boldsymbol{u}_k), \tag{3}$$

where $\boldsymbol{h}(\cdot)$ is a non-linear function, and $\boldsymbol{A}(\cdot)$ is a matrix of functions.

The state-space model of the system (1)–(2) can be expressed as

$$\hat{\boldsymbol{x}}_{k+1} = \boldsymbol{A}(\hat{\boldsymbol{x}}_k)\hat{\boldsymbol{x}}_k + \boldsymbol{h}(\boldsymbol{u}_k), \tag{4}$$

$$\hat{\boldsymbol{y}}_{k+1} = \boldsymbol{C}_{k+1}\hat{\boldsymbol{x}}_{k+1}, \tag{5}$$

where $\hat{\boldsymbol{x}}_{k+1} \in \mathbb{R}^n$ and $\hat{\boldsymbol{y}}_{k+1} \in \mathbb{R}^m$ stand for the state and output estimates, respectively. The problem is to determine $\boldsymbol{A}(\cdot)$, $\boldsymbol{C}_{k+1}$ and $\boldsymbol{h}(\cdot)$, given a set of input-output measurements $\{(\boldsymbol{u}_k, \boldsymbol{y}_k)\}_{k=0}^{n_t-1}$. Moreover, it is assumed that the true state vector $\boldsymbol{x}_k$ is, in particular, unknown. Without loss of generality, it is possible to assume that

$$\boldsymbol{A}(\hat{\boldsymbol{x}}_k) = \text{diag}[a_{1,1}(\hat{\boldsymbol{x}}_k), \ldots, a_{n,n}(\hat{\boldsymbol{x}}_k)]. \tag{6}$$

Thus, the problem reduces to identifying $a_{i,i}(\hat{\boldsymbol{x}}_k)$, $h_i(\boldsymbol{u}_k)$, $i = 1, \ldots, n$, and $\boldsymbol{C}_{k+1}$, i.e., to obtaining $\boldsymbol{A}(\cdot)$ and $\boldsymbol{h}(\cdot)$. Assuming that $|a_{i,i}(\hat{\boldsymbol{x}}_k)| < 1, i = 1, \ldots, n$, it can be shown (Witczak *et al.*, 2002) that the model (4)–(5) is globally asymptotically stable (Witczak *et al.*, 2002). This implies that $a_{i,i}(\hat{\boldsymbol{x}}_k)$ should have the following structure:

$$a_{i,i}(\hat{\boldsymbol{x}}_k) = \tanh(s_{i,i}(\hat{\boldsymbol{x}}_k)), \quad i = 1, \ldots, n, \tag{7}$$

where $\tanh(\cdot)$ is the hyperbolic tangent function, and $s_{i,i}(\hat{\boldsymbol{x}}_k)$ is a function to be determined.

Undoubtedly, many tools can be employed to obtain (4)–(5), e.g., neural networks or Genetic Programming (GP) (Koza, 1992). GP is an extension of genetic algorithms (Michalewicz, 1996), which are a broad class of stochastic optimization algorithms inspired by some biological processes, which allow populations of organisms to adapt to their surrounding environment. The main difference between these two approaches is that in GP the evolving individuals are parse trees rather than fixed-length binary strings. The main advantage of GP over neural networks is that the models resulting from this approach are less sophisticated (from the point of view of the number of parameters).

Since $s_{i,i}(\hat{\boldsymbol{x}}_k)$, $h_i(\boldsymbol{u}_k)$, $i = 1, \ldots, n$, are assumed to be (in general) non-linear functions, they can easily be represented as trees, cf. Fig. 2. The language of trees in GP is formed by user-defined sets of functions $\mathbb{F}$ and terminals $\mathbb{T}$, which form the nodes of the trees (cf. Fig. 2). The functions should be chosen so as to be *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 non-linear systems. In the case of a parameterized tree, as shown in Fig. 2, the terminal set is composed of variables only. Such a parameterization has proven to be especially useful for model designing purposes (Witczak *et al.*, 2002; Witczak, 2003; Witczak and Korbicz, 2004). On the other hand, it leads to the problem of non-linear parameter estimation, which has to be solved by some non-linear programming tools, e.g., the Adaptive Random Search (ARS) algorithm (Walter and Pronzato, 1997).

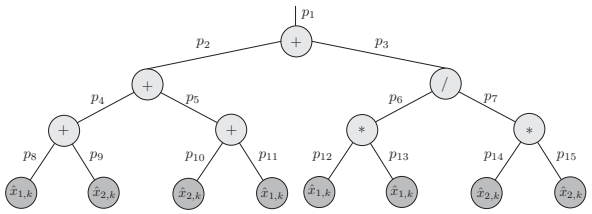


Fig. 2. Exemplary tree representing $s_{i,i}(\hat{\boldsymbol{x}}_k)$.

As a result of applying the above approach to the identification of (1)–(2), each entry of $\boldsymbol{A}(\hat{\boldsymbol{x}}_k)$ and $\boldsymbol{h}(\boldsymbol{u}_k)$ can be obtained with a population of trees evolved by the GP algorithm. It should be pointed out that for that particular purpose two sets of terminals can be distinguished, i.e., one for $\boldsymbol{A}(\hat{\boldsymbol{x}}_k)$ ($\mathbb{T} = \{\hat{\boldsymbol{x}}_k\}$) and the other for $\boldsymbol{h}(\boldsymbol{u}_k)$ ($\mathbb{T} = \{\boldsymbol{u}_k\}$).

As can be observed, parameter estimation involves the computation of $\boldsymbol{C}_k$, which is necessary to obtain the output error $\boldsymbol{\varepsilon}_k$ and, consequently, the value of the fitness function. To tackle this problem, for each trial point $\boldsymbol{p}$ it is necessary to first set an initial state estimate $\hat{\boldsymbol{x}}_0$, and then to obtain the state estimate $\hat{\boldsymbol{x}}_k$, $k = 1, \ldots, n_t - 1$. Knowing the state estimate and using the least-squares method, it is possible to obtain $\boldsymbol{C}_k$ (assuming $\boldsymbol{C}_k = \boldsymbol{C}$) by solving the equation

$$\boldsymbol{C}\sum_{k=0}^{n_t-1}\hat{\boldsymbol{x}}_k\hat{\boldsymbol{x}}_k^T = \sum_{k=0}^{n_t-1}\boldsymbol{y}_k\hat{\boldsymbol{x}}_k^T. \tag{8}$$

It should also be pointed out that the order n of the model is in general unknown and hence should be determined through experiments.

2.2. Observer Design with Genetic Programming

Let us consider the class of non-linear systems described by the following equations:

$$\boldsymbol{x}_{k+1} = g\left(\boldsymbol{x}_k\right) + \boldsymbol{h}(\boldsymbol{u}_k + \boldsymbol{L}_{1,k}\boldsymbol{f}_k) + \boldsymbol{E}_k\boldsymbol{d}_k, \quad (9)$$

$$\boldsymbol{y}_{k+1} = \boldsymbol{C}_{k+1}\boldsymbol{x}_{k+1} + \boldsymbol{L}_{2,k+1}\boldsymbol{f}_{k+1}, \quad (10)$$

where $g\left(\boldsymbol{x}_k\right)$ is assumed to be continuously differentiable with respect to $\boldsymbol{x}_k$, $\boldsymbol{f}_k \in \mathbb{R}^s$ stands for the fault signal, $\boldsymbol{d}_k \in \mathbb{R}^q$ is the unknown input, and $\boldsymbol{L}_{1,k}$, $\boldsymbol{L}_{2,k}$, $\boldsymbol{E}_k$ are their distribution matrices. Similarly to the Extended Kalman Filter (Korbicz *et al.*, 2004), the UIO presented in (Chen and Patton, 1999, pp.98-108) can be extended to the class of non-linear systems (9)–(10). This leads to the following structure of the Extended UIO (EUIO):

$$\hat{\boldsymbol{x}}_{k+1/k} = g\left(\hat{\boldsymbol{x}}_k\right) + \boldsymbol{h}(\boldsymbol{u}_k), \quad (11)$$

$$\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_{k+1/k} + \boldsymbol{H}_{k+1}\boldsymbol{\varepsilon}_{k+1/k} + \boldsymbol{K}_{1,k+1}\boldsymbol{\varepsilon}_k, \quad (12)$$

and

$$\boldsymbol{\varepsilon}_{k+1/k} = \boldsymbol{y}_k - \boldsymbol{C}_{k+1}\hat{\boldsymbol{x}}_{k+1/k}, \quad \boldsymbol{\varepsilon}_k = \boldsymbol{y}_k - \boldsymbol{C}_k\hat{\boldsymbol{x}}_k, \quad (13)$$

where the way of calculating the gain $\boldsymbol{K}_{1,k+1}$ and unknown input decoupling $\boldsymbol{H}_{k+1}$ matrices is given in (Witczak *et al.*, 2002; Witczak, 2003; Witczak and Korbicz, 2004).

It should also be pointed out that the matrix $\boldsymbol{A}_k$ used in the designing procedure is now defined by

$$\boldsymbol{A}_k = \left.\frac{\partial g\left(\boldsymbol{x}_k\right)}{\partial \boldsymbol{x}_k}\right|_{\boldsymbol{x}_k = \hat{\boldsymbol{x}}_k}. \quad (14)$$

The main objective of this section is to show that the convergence of the EUIO strongly depends on an appropriate choice of the instrumental matrices $\boldsymbol{R}_k$ and $\boldsymbol{Q}_k$ (measurement and process noise covariance matrices, respectively, in the stochastic setting). Moreover, the fault-free mode is assumed, i.e., $\boldsymbol{f}_k = \boldsymbol{0}$.

For notational convenience, let us define the *a-priori* state estimation error

$$\boldsymbol{e}_{k+1/k} = \boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1/k}. \quad (15)$$

As usual, to perform further derivations, it is necessary to linearize the model around the current state estimate $\hat{\boldsymbol{x}}_k$. This leads directly to the classical approximation

$$\boldsymbol{e}_{k+1/k} \approx \boldsymbol{A}_k\boldsymbol{e}_k + \boldsymbol{E}_k\boldsymbol{d}_k. \quad (16)$$

In order to avoid the above approximation, the diagonal matrix $\boldsymbol{\alpha}_k = \mathrm{diag}(\alpha_{1,k}, \ldots, \alpha_{n,k})$ can be introduced (Witczak *et al.*, 2002), which makes it possible to establish the following exact equality:

$$\boldsymbol{e}_{k+1/k} = \boldsymbol{\alpha}_k\boldsymbol{A}_k\boldsymbol{e}_k + \boldsymbol{E}_k\boldsymbol{d}_k. \quad (17)$$

The problem is to obtain an appropriate form of the instrumental matrices $\boldsymbol{Q}_{k-1}$ and $\boldsymbol{R}_k$ in such a way as to ensure the convergence of the observer or to adequately maximize the bounds of the diagonal elements of the matrix $\boldsymbol{\alpha}_k$.

For that purpose, Witczak *et al.* (2002) performed a comprehensive convergence analysis with the Lyapunov method. As a result, they obtained the following conditions:

$$\bar{\sigma}\left(\boldsymbol{\alpha}_k\right) \le \gamma_1 = \frac{\underline{\sigma}\left(\boldsymbol{A}_k\right)}{\bar{\sigma}\left(\boldsymbol{A}_k\right)}\left(\frac{(1-\zeta)\underline{\sigma}\left(\boldsymbol{P}_k\right)}{\bar{\sigma}\left(\boldsymbol{A}_{1,k}\boldsymbol{P}_k'\boldsymbol{A}_{1,k}^T\right)}\right)^{\frac{1}{2}},$$

and

$$\bar{\sigma}\left(\boldsymbol{\alpha}_k - \boldsymbol{I}\right) \le \gamma_2 = \frac{\underline{\sigma}\left(\boldsymbol{A}_k\right)}{\bar{\sigma}\left(\boldsymbol{A}_k\right)} \times \left(\frac{\underline{\sigma}\left(\boldsymbol{C}_k^T\right)\underline{\sigma}\left(\boldsymbol{C}_k\right)}{\bar{\sigma}\left(\boldsymbol{C}_k^T\right)\bar{\sigma}\left(\boldsymbol{C}_k\right)}\frac{\underline{\sigma}\left(\boldsymbol{R}_k\right)}{\bar{\sigma}\left(\boldsymbol{C}_k\boldsymbol{P}_k\boldsymbol{C}_k^T + \boldsymbol{R}_k\right)}\right)^{\frac{1}{2}}.$$

Bearing in mind the fact that $\boldsymbol{\alpha}_k$ is a diagonal matrix, the above inequalities can be expressed as

$$\max_{i=1,\ldots,n}|\alpha_{i,k}| \le \gamma_1, \quad \max_{i=1,\ldots,n}|\alpha_{i,k} - 1| \le \gamma_2. \quad (18)$$

Since (cf. Chen and Patton, 1999, pp. 98–108):

$$\boldsymbol{P}_k = \boldsymbol{A}_{1,k}\boldsymbol{P}_k'\boldsymbol{A}_{1,k}^T + \boldsymbol{T}_k\boldsymbol{Q}_{k-1}\boldsymbol{T}_k^T + \boldsymbol{H}_k\boldsymbol{R}_k\boldsymbol{H}_k^T, \quad (19)$$

it is clear that an appropriate selection of the instrumental matrices $\boldsymbol{Q}_{k-1}$ and $\boldsymbol{R}_k$ may enlarge the bounds γ_1 and γ_2 and, consequently, the domain of attraction. Indeed, if the conditions (18) are satisfied, then $\hat{\boldsymbol{x}}_k$ converges to $\boldsymbol{x}_k$.

Unfortunately, analytical derivation of the matrices $\boldsymbol{Q}_{k-1}$ and $\boldsymbol{R}_k$ seems to be an extremely difficult problem. However, it is possible to set the above matrices as follows: $\boldsymbol{Q}_{k-1} = \beta_1\boldsymbol{I}$, $\boldsymbol{R}_k = \beta_1\boldsymbol{I}$, with β_1 and β_1 large enough. On the other hand, it is well known that the convergence rate of such an EKF-like approach can be increased by an appropriate selection of the covariance matrices $\boldsymbol{Q}_{k-1}$ and $\boldsymbol{R}_k$, i.e., the more accurate (near "true" values) the covariance matrices, the better the convergence rate. This means that in the deterministic case ($\widetilde{k} = \boldsymbol{0}$ and $\boldsymbol{v}_k = \boldsymbol{0}$), both matrices should be zero. Unfortunately, such an approach usually leads to the divergence of the observer as well as other computational problems. To tackle this issue, a compromise between the convergence and the convergence rate should be established. This can easily be done by setting the instrumental matrices as

$$\boldsymbol{Q}_{k-1} = \beta_1\boldsymbol{\varepsilon}_{k-1}^T\boldsymbol{\varepsilon}_{k-1}\boldsymbol{I} + \delta_1\boldsymbol{I}, \quad (20)$$

$$\boldsymbol{R}_k = \beta_2\boldsymbol{\varepsilon}_k^T\boldsymbol{\varepsilon}_k\boldsymbol{I} + \delta_2\boldsymbol{I}, \quad (21)$$

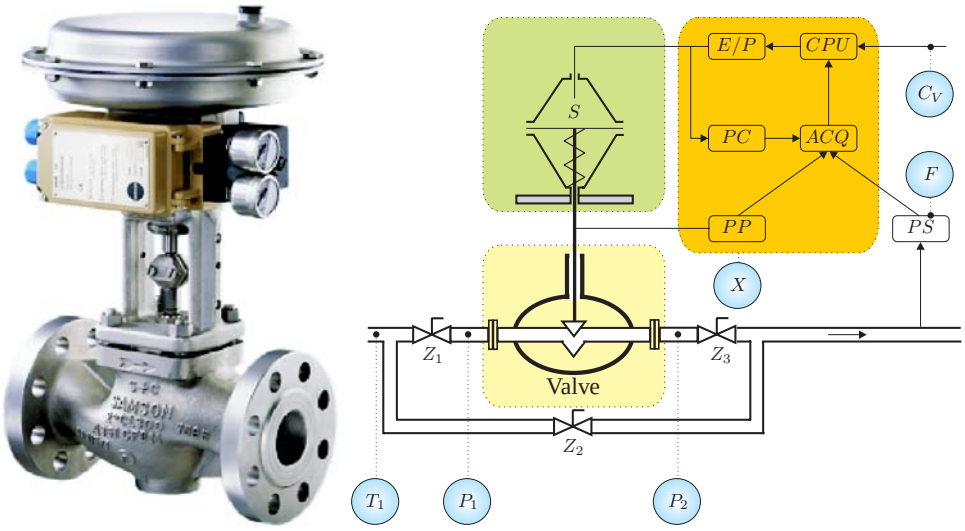


Fig. 3. Actuator and its scheme.

with β_1, β_2 large enough, and δ_1, δ_2 small enough. Although this approach is very simple, it is possible to further increase the convergence rate. Indeed, the instrumental matrices can be set as

$$\boldsymbol{Q}_{k-1} = q^2(\varepsilon_{k-1})\boldsymbol{I} + \delta_1\boldsymbol{I},$$
$$\boldsymbol{R}_k = r^2(\varepsilon_k)\boldsymbol{I} + \delta_2\boldsymbol{I}, \tag{22}$$

where $q(\varepsilon_{k-1})$ and $r(\varepsilon_k)$ are non-linear functions of the output error ε_k (the squares are used to ensure the positive definiteness of $\boldsymbol{Q}_{k-1}$ and $\boldsymbol{R}_k$). Thus, the problem reduces to identifying the above functions. In particular, in (Witczak *et al.*, 2002) it is shown how to reduce the task of the designing of $\boldsymbol{Q}_{k-1}$ and $\boldsymbol{R}_k$ to the multi-objective and global structure optimization problem of $q(\varepsilon_{k-1})$ and $r(\varepsilon_k)$. Genetic programming is utilized (Witczak *et al.*, 2002; Witczak, 2003; Witczak and Korbicz, 2004) to tackle this problem.

2.3. An Illustrative Example – the DAMADICS Benchmark

DAMADICS (*Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems*) was a research project focused on drawing together wide-ranging techniques and fault diagnosis within the framework of a real application to on-line diagnosis of a 5-stage evaporisation plant of a sugar factory in Lublin, Poland. The project was focused on the diagnosis of valve (cf. Fig. 3) plant actuators and looked towards real implementation methods for new actuator systems. The sugar factory was a subcontractor (under the Warsaw University of Technology) providing real process data and the evaluation of trials of fault diagnosis methods.

The control valve constitutes the means used to prevent, permit and/or limit the flow of sugar juice through the control system (a detailed description of this actuator can be found in (DAMADICS, 2004)). As can be seen in Fig. 3, the following process variables can be measured: CV is the control signal, $P1$ is the pressure at the inlet of the valve, $P2$ is the pressure at the outlet of the valve, $T1$ is the juice temperature at the inlet of the valve, X is the servomotor rod displacement, F is the juice flow at the outlet of the valve. In Fig. 3, three additional bypass valves (denoted by z_1, z_2, and z_3) can be seen. The state of these valves can be controlled manually by the operator. They are introduced for manual process operation, actuator maintenance and safety purposes. The data gathered from the real plant are available on the DAMADICS website (DAMADICS, 2004). Although a large amount of real data are available, they do not cover all faulty situations.

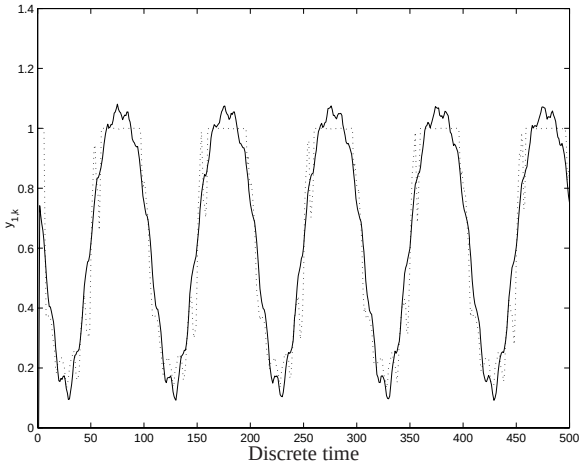
The objective of this section is to design the state-space model of the actuator being considered (cf. Fig. 3) according to the approach described in Section 2.1. The parameters used during the identification process were $n_m = 200$, $n_d = 10$, $n_s = 10$, $\mathbb{F} = \{+, *, /\}$. For the sake of comparison, the linear state-space model was obtained with the use of the MATLAB System Identification Toolbox. In both the linear and non-linear cases, the order of the model was tested between $n = 2, \ldots, 8$. Unfortunately, the relation between the input $\boldsymbol{u}_k = (CV, P1, P2, T1)$ and the juice flow $y_{1,k}$ ($\boldsymbol{y}_k = (F, X)$) cannot be modelled by a linear state-space model. Indeed, the modelling error was approximately 35%, thus making the linear model unacceptable. On the other hand, the relation between the input $\boldsymbol{u}_k$ and the rod displacement $y_{2,k}$ can be modelled, with very good results, by the linear

state-space model. Bearing this in mind, the identification process was decomposed into two phases, i.e.,

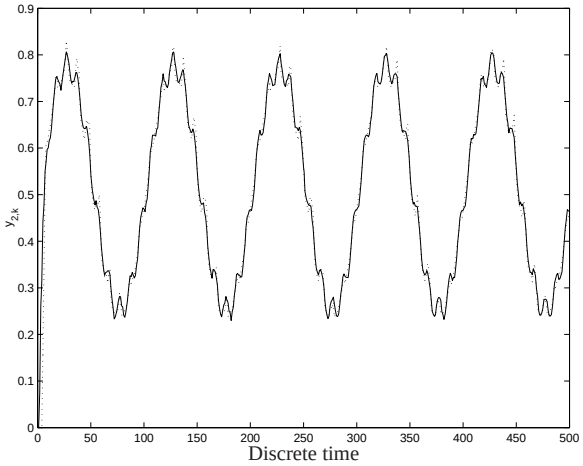
1. Derivation of a relation between the rod displacement and the input with a linear state-space model.

2. Derivation of a relation between the juice flow and the input with a non-linear state-space model designed by GP.

Experimental results showed that the best-suited linear model is of the order $n = 2$. After 50 runs of the GP algorithm performed for each model order, it was found that the order of the model which provides the best approximation quality is $n = 2$. The mean-squared output error for the obtained model was 0.0079 (the model structure can be found in (Witczak and Korbicz, 2004)). The response of the model obtained for the validation data set is given in Fig. 4.



(a)



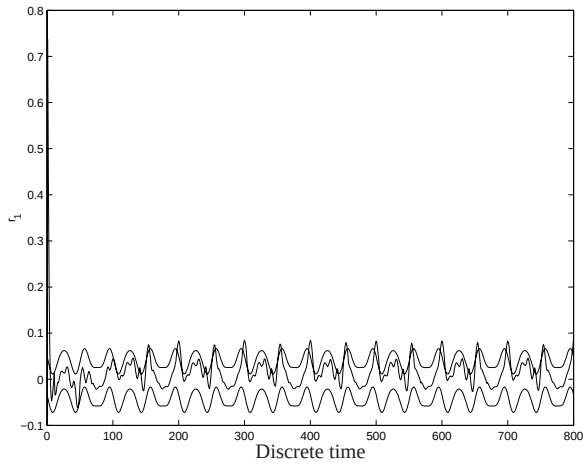
(b)

Fig. 4. System (dotted line) and model (solid line) outputs (juice flow (a), rod displacement (b)) for the validation data set.

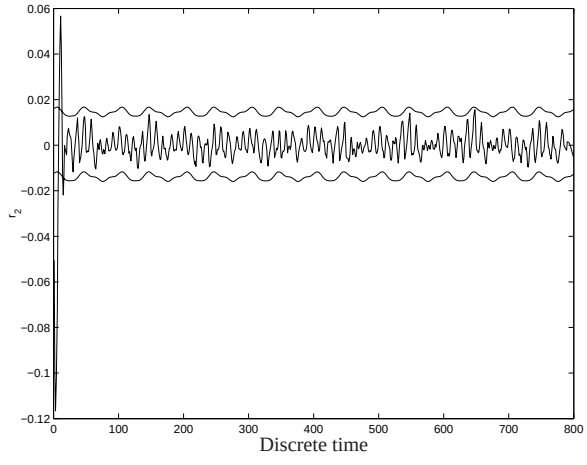
The main differences between the behaviour of the model and the system were observed for the non-linear model (juice flow) during system saturation. This inaccuracy constitutes the main part of modelling uncertainty.

Since the state-space model is given, it is possible to design the EUIO presented in Section 2.2. To tackle the determination of the unknown input distribution matrix $\boldsymbol{E}_k$, the approach proposed in (Witczak and Korbicz, 2004) was employed, and then the approach of Section 2.2 along with the threshold selector described in (Witczak and Korbicz, 2004) were used for fault detection. This threshold selector is based on a polynomial describing the relation between the control signal and the residual. In particular, adaptive residual bounds are provided based on the parameter confidence region of the above polynomial.

Since the method of designing an appropriate threshold is known, it is possible to check fault detection capabilities of the presented observer-based fault detection scheme. Figure 5 presents the results of fault detection for



(a)



(b)

Fig. 5. Residuals for the flow sensor fault and its bounds (juice flow – (a), rod displacement – (b)).

the flow sensor fault (with a small magnitude). As can be observed, the fault can be detected with the use of the juice flow residual. A complete report regarding fault diagnosis of the DAMADICS benchmark can be found in (Witczak and Korbicz, 2004).

3. Neural Networks in FDI

Generally, neural networks (Gupta *et al.*, 2003) can be perceived as a conveniently parameterized set of non-linear maps. In the last fifteen years, neural networks have been successfully used in solving complex problems in modelling and pattern recognition (see Gupta *et al.*, 2003 and the references therein). In the case of pattern recognition, a finite set of input-output pairs is given, where the inputs represents the objects to be recognised while the outputs stand for the pattern classes to which they belong. Thus, the role of a neural network is to approximate the map between these two spaces. In the case of modelling, it is assumed that the input-output relation is formed by a non-linear system, and the role of a neural network is to approximate the behaviour of this system. In both cases, the application of neural networks is justified by the assumption that there exists a non-linear input-output map. A key theoretical result behind both applications is the fact that neural networks are universal approximators (Gupta *et al.*, 2003). There are, of course, many different properties (see, e.g. (Chandra and Sing, 2004)), which make neural networks attractive for practical applications.

At the beginning of the 1990s, neural networks were proposed for identification and control (see, e.g., (Narendra, 1990)). A rapid development concerning applications of neural networks to control engineering resulted in a large number of publications related to this subject. Hunt *et al.* (1992) confirmed the fast development of this research area by publishing a survey on neural networks in control engineering. In 1995, a similar work was published by Sjoberg *et al.* (1995) in the context of system identification with neural networks. Nowadays, the number of applications has increased significantly. Fault diagnosis constitutes one of the thrusts of the research effort on neural networks for control (Korbicz *et al.*, 2004).

The main objective of the subsequent part of this paper is to present the development of this particular research area. Rather than providing an exhaustive survey on neural networks in fault diagnosis, the aim is to provide a comprehensive account of the published works that exploit the special nature of neural networks. Indeed, it is impossible to count all publications on fault diagnosis in which neural networks are used as models of the systems being diagnosed. The strategy underlying such an approach boils down to generating the residual with the system and neural network outputs, respectively. Examples of using such an approach with the classical multi-layer perceptron are: leakage detection in an electro-hydraulic cylinder drive in a fluid power system (Watton and Pham, 1997), the diagnosis of non-catastrophic faults in a nuclear plant (Weerasinghe *et al.*, 1998), and process valve actuator fault diagnosis (Karpenko *et al.*, 2003). Similar examples relating to dynamic neural networks are: the diagnosis of a chemical plant (Fuente and Saludes, 2000), the diagnosis of a valve actuator (Korbicz *et al.*, 2004; Patan and Parisini, 2005), and the diagnosis of a steam evaporator (Janczak, 2005).

There are a number of works concerning observer design with neural networks (Alessandri *et al.*, 1997; Guo and Zhu, 2002). Thus, if non-linear state-space models are available, then these approaches can be utilized for residual generation and fault diagnosis. Moreover, robustness with respect to model uncertainty can also be achieved by using the concept of an unknown input. Unfortunately, when the direction of faults is similar to that of an unknown input, then the unknown input decoupling procedure may considerably impair the fault sensitivity. If the above-mentioned approach fails, then describing model uncertainty in a different way seems to be a good remedy. One of possible approaches is to use statistical techniques (Atkinson and Donev, 1992; Walter and Pronzato, 1997) (for an example regarding different approaches, the reader is referred to (Delebecque *et al.*, 2003)) to obtain parameter uncertainty of the model and, consequently, model output uncertainty. Such parameter uncertainty is defined as the parameter confidence region (Atkinson and Donev, 1992; Walter and Pronzato, 1997) containing a set of admissible parameters that are consistent with the measured data. Thus it is evident that parameter uncertainty depends on measurement uncertainty, i.e., noise, disturbances, etc.

Knowledge about parameter uncertainty makes it possible to design the so-called adaptive threshold (Frank *et al.*, 1999). The adaptive threshold, contrary to the fixed one (cf. Fig. 6), bounds the residual at a level that is dependent on model uncertainty, and hence it provides more reliable fault detection. Contrary to the typical industrial applications of neural networks that are presented in the literature (Chen and Patton, 1999; Karpenko *et al.*, 2003; Korbicz *et al.*, 2004), Witczak *et al.* (2006) defined the task of designing a neural network in such a way as to obtain a model with possibly small uncertainty. Indeed, the approaches presented in the literature try to obtain a model that is best suited to a particular data set. This may result in a model with relatively large uncertainty. Degraded performance of fault diagnosis constitutes a direct consequence of using such models. To tackle this challenging problem, the GMDH (Group Method of Data Handling) approach was adapted and modified (Ivakhnenko and Mueller, 1995; Korbicz *et*

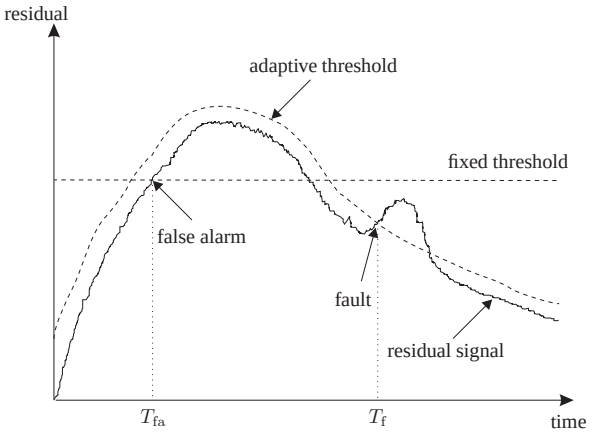


Fig. 6. Principle of an adaptive threshold.

al., 2004). They proposed a complete design procedure concerning the application of GMDH neural networks to robust fault detection. Starting from a set of input-output measurements of the system, it is shown how to estimate the parameters and the corresponding uncertainty of a neuron using the so-called bounded-error approach (Milanese *et al.*, 1996; Walter and Pronzato, 1997). As a result, a tool that is able to generate an adaptive threshold was obtained. The methodology developed for parameter and uncertainty estimation of a neuron makes it possible to formulate an algorithm that allows obtaining a neural network with relatively small modelling uncertainty. All the hard computations regarding the design of the GMDH neural network are performed off-line and hence the problem regarding the time consuming calculations is not of paramount importance. The approach can also be extended for dynamic systems by using the dynamic neuron structure (Mrugalski and Witczak, 2002).

It is well known that the reliability of such fault diagnosis schemes is strongly dependent on model uncertainty, i.e., the mismatch between a neural network and the system being considered. Thus, it is natural to minimize model uncertainty as far as possible. This can be realized with the application of Optimum Experimental Design (OED) theory (Atkinson and Donev, 1992; Uciński, 2005; Walter and Pronzato, 1997). Some authors have conducted active investigations in this important research area. White (1989), MacKay (1992), and Cohn (1994) showed the attractiveness of the application of OED to neural networks. Fukumizu (1996; 2000) developed the so-called statistical active learning technique, which is based on the general theory of OED. Recently, Witczak and Prętki (2005) developed a D-optimum experimental design strategy that can be used for training single-output neural networks. They also showed how to use the obtained network for robust fault detection with an adaptive threshold. In (Witczak, 2006), the author showed how to extend this technique to multi-input multi-output neural networks. He also proposed a sequential experimental design algorithm that allows obtaining a one-step-ahead D-optimum input. This algorithm can be perceived as a hybrid one since it can be used for both training and data development. Section 3.1 presents selected parts of the above-described design methodology.

Finally, there are also a large number of approaches that use neural networks as pattern classifiers (Korbicz *et al.*, 2004) to tackle the FDI problem. Instead of using neural networks as models of the systems being diagnosed, the networks are trained to recognise different modes of the system, i.e., both faulty and non-faulty ones. Examples of using such an approach are: FDI in hydraulic fluid power systems (Le *et al.*, 1997; Le *et al.*, 1998), FDI in machine dynamics and vibration problems (Yen and Lin, 2000), sensor fault diagnosis (Zhang *et al.*, 2000), fault diagnosis of chemical processes (Zhao *et al.*, 1998), and fault diagnosis of a two-tank system (Korbicz *et al.*, 2004).

3.1. Experimental Design for Neural Networks

Consider a feed-forward neural network given by the following equation (Witczak, 2006):

$$\boldsymbol{y}_{m,k} = \boldsymbol{P}^{(l)} \boldsymbol{g}\left(\boldsymbol{P}^{(n)} \boldsymbol{u}_k\right), \tag{23}$$

where $\boldsymbol{y}_{m,k} \in \mathbb{R}^m$ stands for the model output, $\boldsymbol{g}(\cdot) = [g_1(\cdot), \ldots, g_{n_h}(\cdot), 1]^T$, where $g_i(\cdot) = g(\cdot)$ is a non-linear differentiable activation function,

$$\boldsymbol{P}^{(l)} = \begin{bmatrix} \boldsymbol{p}^{(l)}(1)^T \\ \vdots \\ \boldsymbol{p}^{(l)}(m)^T \end{bmatrix}, \quad \boldsymbol{P}^{(n)} = \begin{bmatrix} \boldsymbol{p}^{(n)}(1)^T \\ \vdots \\ \boldsymbol{p}^{(n)}(n_h)^T \end{bmatrix}, \tag{24}$$

are matrices representing the parameters (weights) of the model, n_h is the number of neurons in the hidden layer. Moreover, $\boldsymbol{u}_k \in \mathbb{R}^{n_r+1}$, $\boldsymbol{u}_k = [u_{1,k}, \ldots, u_{n_r,k}, 1]^T$ where $u_{i,k}$, $i = 1, \ldots, n_r$ are system inputs. For the sake of notational simplicity, define the following parameter vector:

$$\boldsymbol{p} = \big[\boldsymbol{p}^{(l)}(1)^T, \ldots, \boldsymbol{p}^{(l)}(m)^T, \boldsymbol{p}^{(n)}(1)^T, \ldots, \boldsymbol{p}^{(n)}(n_h)^T\big]^T,$$

where $\boldsymbol{p} \in \mathbb{R}^{n_p}$, $n_p = m(n_h+1) + n_h(n_r+1)$. Consequently, (23) can be written in a more compact form:

$$\boldsymbol{y}_{m,k} = \boldsymbol{f}\left(\boldsymbol{p}, \boldsymbol{u}_k\right), \tag{25}$$

where $\boldsymbol{f}(\cdot)$ is a non-linear function representing the structure of the neural network.

$$\frac{\partial \boldsymbol{f}\left(\boldsymbol{p},\boldsymbol{u}_k\right)}{\partial \boldsymbol{p}}=\begin{bmatrix} \boldsymbol{g}\left(\boldsymbol{P}^{(n)}\boldsymbol{u}_k\right)^T & \boldsymbol{0}^T_{(m-1)(n_h+1)} & p^l_1(1)g'\left(\boldsymbol{u}_k^T p^n(1)\right)\boldsymbol{u}_k^T & \dots & p^l_{n_h}(1)g'\left(\boldsymbol{u}_k^T p^n(n_h)\right)\boldsymbol{u}_k^T \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{0}^T_{(m-1)(n_h+1)} & \boldsymbol{g}\left(\boldsymbol{P}^{(n)}\boldsymbol{u}_k\right)^T & p^l_1(m)g'\left(\boldsymbol{u}_k^T p^n(1)\right)\boldsymbol{u}_k^T & \dots & p^l_{n_h}(m)g'\left(\boldsymbol{u}_k^T p^n(n_h)\right)\boldsymbol{u}_k^T \end{bmatrix}. \quad (30)$$

Assume that the system output satisfies

$$\boldsymbol{y}_k=\boldsymbol{y}_{m,k}+\boldsymbol{\epsilon}_k=\boldsymbol{f}\left(\boldsymbol{p},\boldsymbol{u}_k\right)+\boldsymbol{\epsilon}_k, \quad (26)$$

where the noise $\boldsymbol{\epsilon}$ is zero-mean, Gaussian, and uncorrelated in k, i.e., its statistics are

$$\mathcal{E}(\boldsymbol{\epsilon}_k)=\boldsymbol{0}, \quad \mathcal{E}(\boldsymbol{\epsilon}_i\boldsymbol{\epsilon}_k^T)=\delta_{i,k}\boldsymbol{C}, \quad (27)$$

where $\boldsymbol{C}\in\mathbb{R}^{m\times m}$ is a known positive-definite matrix of the form $\boldsymbol{C}=\sigma^2\boldsymbol{I}_m$, and σ^2 and $\delta_{i,k}$ stand for the variance and Kronecker's delta symbol, respectively. Under such an assumption, the theory of experimental design (Atkinson and Donev, 1992; Walter and Pronzato, 1997) can be exploited to develop a suitable training data set that allows obtaining a neural network with considerably smaller uncertainty than those designed without it. First, let us define the so-called Fisher information matrix, which constitutes a measure of parametric uncertainty of (23):

$$\boldsymbol{P}^{-1}=\sum_{k=1}^{n_t}\boldsymbol{R}_k\boldsymbol{R}_k^T, \quad (28)$$

$$\boldsymbol{R}_k=\left(\frac{\partial \boldsymbol{f}\left(\boldsymbol{p},\boldsymbol{u}_k\right)}{\partial \boldsymbol{p}}\right)^T_{\boldsymbol{p}=\hat{\boldsymbol{p}}}, \quad (29)$$

and the Jacobi matrix is defined by Eqn. (30), where $g'(t)=\mathrm{d}g(t)/\mathrm{d}t$, $\hat{\boldsymbol{p}}$ is the least-squares estimate of $\boldsymbol{p}$, and n_t stands for the number of input-output measurements. It is easy to observe that the FIM (28) depends on the experimental conditions $\xi=[\boldsymbol{u}_1,\dots,\boldsymbol{u}_{n_t}]$. Thus, optimal experimental conditions can be found by choosing $\boldsymbol{u}_k$, $k=1,\dots,n_t$, so as to minimize some scalar function of (28). Such a function can be defined in various ways (Fukumizu, 2000; Uciński, 2005), and here are the most popular of them:

- D-optimality criterion:

$$\Phi(\xi)=\det\boldsymbol{P}, \quad (31)$$

- G-optimality criterion:

$$\Phi(\xi)=\max_{\boldsymbol{u}_k\in\mathbb{U}}\phi(\xi,\boldsymbol{u}_k). \quad (32)$$

where $\mathbb{U}$ stands for the set of admissible $\boldsymbol{u}_k$ that can be used for the system being considered (design space), and

$$\phi(\xi,\boldsymbol{u}_k)=\mathrm{trace}\left(\boldsymbol{R}_k^T\boldsymbol{P}\boldsymbol{R}_k\right)=\sum_{i=1}^{m}\boldsymbol{r}_{i,k}\boldsymbol{P}\boldsymbol{r}_{i,k}^T, \quad (33)$$

where $\boldsymbol{r}_{i,k}$ stands for the i-th row of $\boldsymbol{R}_k^T$.

A valuable property of the FIM is that its inverse constitutes an approximation of the covariance matrix for $\hat{\boldsymbol{p}}$ (Goodwin and Payne, 1977). Thus, D-optimum design minimizes the volume of the confidence ellipsoid approximating the feasible parameter set of (23) (see, e.g., (Atkinson and Donev, 1992, Sec. 6.2) for further explanations). G-optimum design minimizes the variance of the estimated response of (23). The D-optimality criterion has been employed by many authors in the development of computer algorithms for calculating optimal experimental designs. Another important property is that D-optimum designs are invariant with respect to non-degenerate linear transformations of the model. It is also important to underline that, from the practical point of view, D-optimum designs often perform well according to other criteria (see (Atkinson and Donev, 1992) and the references therein for more details). For further explanations regarding D-optimality criteria, the reader is referred to the excellent textbooks (Atkinson and Donev, 1992; Fedorov and Hackl, 1997; Uciński, 2005; Walter and Pronzato, 1997).

Since fault diagnosis applications are the primary purpose, the main objective is to use a design criterion which makes it possible to obtain accurate bounds of the system output (cf. Fig. 7). Indeed, it is rather pointless to assume that it is possible to develop a neural network with arbitrarily small uncertainty, i.e., to obtain a perfect model of the system. A more realistic task is to design a model that will provide reliable knowledge about the bounds of the system output that reflect the expected system behaviour. The design methodology of such robust techniques rests on the paradigm that fault diagnosis and control schemes should perform reliably for all kinds of system behaviour that are consistent with output bounds. This is in contradiction with the conventional approaches, where fault diagnosis and control schemes are designed to be optimal for one single model.

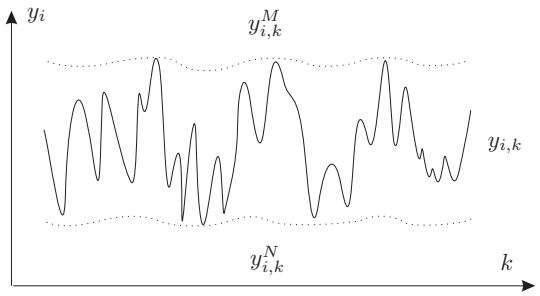


Fig. 7. i-th output of the system and its bounds obtained with a neural network.

The bounds presented in Fig. 7 can be described as follows:

$$y_{i,k}^N \leq y_{i,k} \leq y_{i,k}^M, \quad i = 1, \ldots, m. \tag{34}$$

In (Chryssolouris *et al.*, 1996), the authors developed an approach that can be used for determining (34) (that forms the $100(1-\alpha)$ confidence interval of $y_{i,k}$) for single output ($m = 1$) neural networks. In (Witczak, 2006), the approach of (Chryssolouris *et al.*, 1996) was extended to multi-output models, i.e., it can be shown that $y_{i,k}^N$ and $y_{i,k}^M$ (that form the $100(1-\alpha)$ confidence interval of $y_{i,k}$) can be approximated as follows:

$$y_{i,k}^N = \hat{y}_{i,k} - t_{n_t-n_p}^{\alpha/2} \hat{\sigma} \left(1 + \boldsymbol{r}_{i,k} \boldsymbol{P} \boldsymbol{r}_{i,k}^T\right)^{1/2}, \tag{35}$$

$$y_{i,k}^M = \hat{y}_{i,k} + t_{n_t-n_p}^{\alpha/2} \hat{\sigma} \left(1 + \boldsymbol{r}_{i,k} \boldsymbol{P} \boldsymbol{r}_{i,k}^T\right)^{1/2}, \tag{36}$$

$i = 1, \ldots, m$, where $t_{n_t-n_p}^{\alpha/2}$ is the t-Student distribution quantile, and $\hat{\sigma}$ is the standard deviation estimate. Bearing in mind the fact that the primary purpose of this work is to develop reliable bounds of the system output, it is clear from (34), (35), and (36) that the G-optimality criterion should be selected.

When some experiments are repeated, the number n_e of distinct $\boldsymbol{u}_k$s is smaller than the total number of observations n_t. The design resulting from this approach is called the *continuous experimental design* and can be described as follows:

$$\xi = \left\{ \begin{array}{cccc} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \ldots & \boldsymbol{u}_{n_e} \\ \mu_1 & \mu_2 & \ldots & \mu_{n_e} \end{array} \right\}, \tag{37}$$

where $\boldsymbol{u}_k$s are said to be the *support points*, and $\mu_1, \ldots, \mu_{n_e}, \mu_k \in [0, 1]$ are their weights, which satisfy $\sum_{k=1}^{n_e} \mu_k = 1$.

Thus, the Fisher information matrix can now be defined as follows:

$$\boldsymbol{P}^{-1} = \sum_{k=1}^{n_e} \mu_k \boldsymbol{R}_k \boldsymbol{R}_k^T. \tag{38}$$

The fundamental property of continuous experimental design is the fact that optimum designs resulting from the D- and G-optimality criteria are the same (the Kiefer-Wolfowitz equivalence theorem Atkinson and Donev, 1992; Walter and Pronzato, 1997; Uciński, 2005). Finally, in the light of this theorem, the design (37) is D-optimum when

$$\phi(\xi, \boldsymbol{u}_k) = \text{trace}\left(\boldsymbol{R}_k^T \boldsymbol{P} \boldsymbol{R}_k\right) \leq n_p, \tag{39}$$

where the equality holds for measurements described by (37).

Generally, it is impossible to obtain the design (37) in an analytical way. To tackle this problem, in (Witczak and Prętki, 2005) the authors employed the Wynn-Fedorov algorithm (Uciński, 2005; Walter and Pronzato, 1997) to obtain a D-optimum design for a single-output neural network, while in (Witczak, 2006) it is shown how to use it for a general multi-output multi-input neural network. The subsequent section presents an illustrative example of using the above approach for fault detection of a valve actuator (DAMADICS benchmark).

3.2. Experimental Results

Let us reconsider an example presented in Section 2.3. The main objective of the subsequent part of this section is to develop a neural network that can be used for fault detection of an industrial valve actuator. The above task was divided into the following steps (Witczak, 2006):

Step 1: Training of a network based on the nominal data set.

Step 2: Design of the experiment with the Wynn-Fedorov algorithm described in (Witczak, 2006) based on the network obtained in *Step 1*.

Step 3: Training of a network based on the data obtained with optimal experimental design.

Based on the experience with an industrial valve actuator, it was observed that the following subset of measured variables is sufficient for fault detection purposes: $\boldsymbol{u} = (CV, P1, 1)$, $y = F$.

In Step 1, a number of experiments (the training of a neural network with the Levenberg-Marquardt algorithm (Walter and Pronzato, 1997)) were performed in order to find a suitable number of hidden neurons n_h (cf. (23)). For that purpose, $n_t = 100$ data points were generated, for which inputs were uniformly spread within the design region $\mathbb{U}$, where $0.25 < u_1 < 0.75$ and $0.6625 < u_2 < 0.8375$. As a result, a neural model consisting of $n_h = 5$ hidden neurons was obtained. The main objective of Step 2 was to utilize the above model

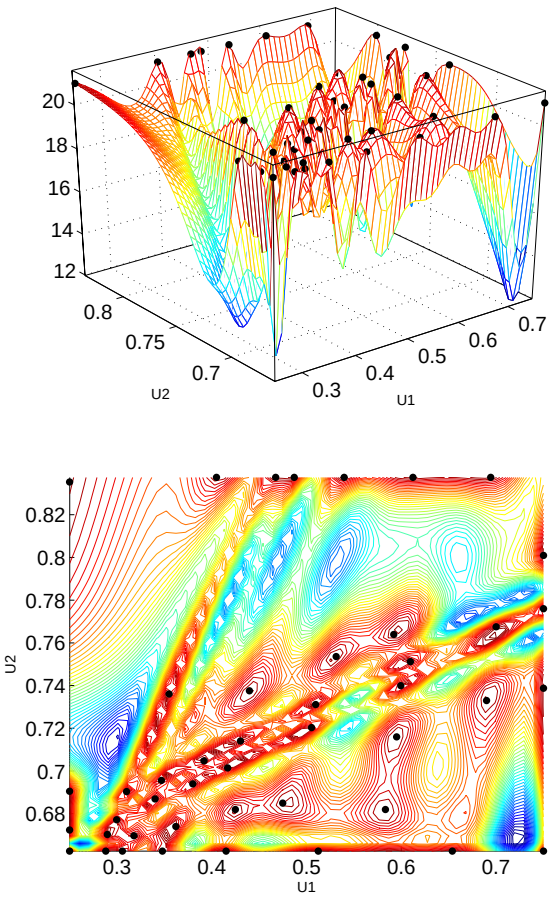


Fig. 8. Variance function and the corresponding support points.

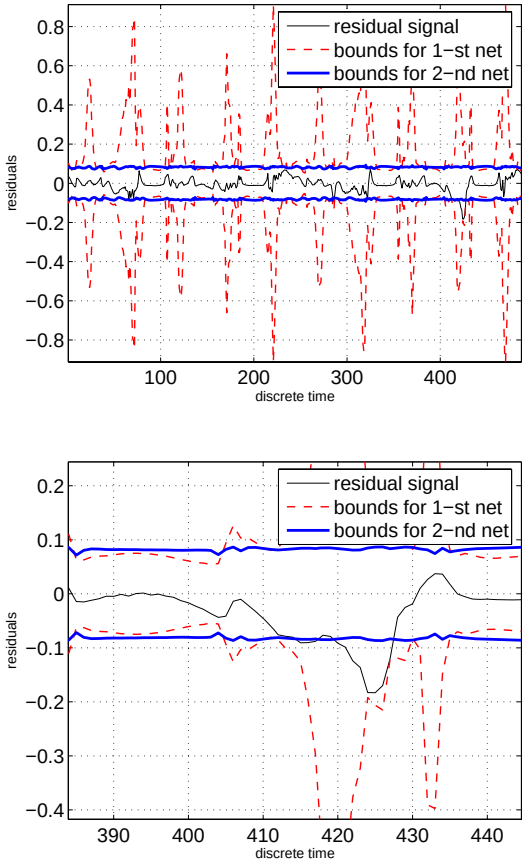


Fig. 9. Residual and adaptive thresholds for the valve clogging fault.

and the Wynn-Fedorov algorithm in order to obtain D-optimum experimental conditions. First, an initial experiment was generated in such a way as to equally divide the design space of $\boldsymbol{u}$. Finally, the Wynn-Fedorov algorithm was applied. Figure 8 shows the support points ($n_e = 45$) and the variance function (33) for the obtained D-optimum design. As can be observed, the D-optimality condition (39) is satisfied. Based on the obtained continuous design, a set consisting of $n_t = 100$ points was obtained and used for data generation. The number of repetitions of each optimal input $\boldsymbol{u}_k$ was calculated by suitably rounding the numbers $\mu_k n_t,\ k = 1, \ldots, n_e$ (Atkinson and Donev, 1992; Walter and Pronzato, 1997). It should be strongly stressed that the data were collected in a steady state because the utilized model (23) was static. Finally, the new data set was used for training the network with the Levenberg-Marquardt algorithm. Figure 9 presents the residual signal obtained with a network trained with the D-optimum data set as well as an adaptive threshold provided by this network (2-nd network). This figure also presents an adaptive threshold provided by a network (1-st network) trained with the data set generated by evenly dividing the design space. It can be observed that the neural network obtained with the use of D-optimum experimental design makes it possible to obtain more accurate bounds than those obtained with a neural network trained in a different manner. Indeed, as can be seen in Fig. 9, the valve clogging fault (which, in the light of its nature, is hard to detect) can be detected with the help of the second network while it is impossible to detect it with the use of the first one. It should be strongly underlined that the situation is even worse when the first network is used for residual generation, i.e., in the presented example it was used for adaptive threshold generation only. Finally, Fig. 10 presents exemplary residuals for flow sensor faults (cf. the results of Section 2.3). A complete report regarding fault diagnosis of the DAMADICS benchmark can be found in (Witczak, 2006).

4. Concluding Remarks

The rapid development of modern fault diagnosis exposes the disadvantages of the classical model-based techniques,

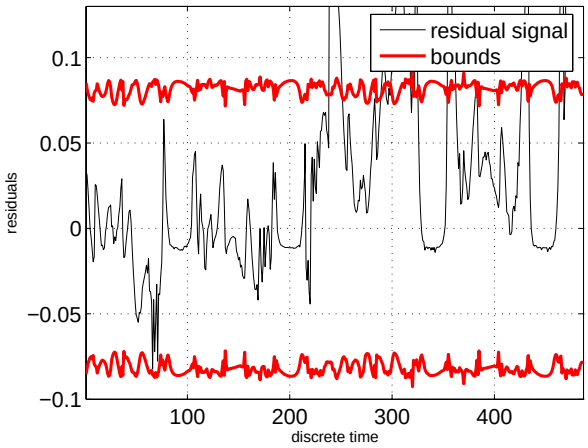


Fig. 10. Residual and adaptive thresholds for the flow sensor fault.

which can be perceived from various angles. In spite of the fact that a large spectrum of analytical techniques for FDI of non-linear systems can be found in the literature, they usually all suffer from the lack of an appropriate mathematical description of the system being considered. Another difficulty is the fact that the classical solutions to the robustness problem may impair the efficiency and quality of fault diagnosis, e.g., when the direction of faults is similar to that of an unknown input. As was shown in the paper, to settle such a challenging problem, it is possible to employ evolutionary algorithms and neural networks. Indeed, a large amount of knowledge on using these techniques for model-based fault diagnosis has been accumulated through the literature since the beginning of the 1990s. One objective of this paper was to summarize the advances in this interesting research area. Rather than providing an exhaustive survey on evolutionary algorithms and neural networks in fault diagnosis, the paper was aimed at providing a comprehensive account of the published works that exploit the special nature of these techniques. Special attention was paid to techniques that integrate the classical and soft computing approaches in such a way as to tackle inevitable non-linearity and robustness problems. In particular, it was shown that the integration of genetic programming and unknown input observers results in a hybrid fault diagnosis technique that is superior over the classical one. Another example concerned the problem of robust fault detection using neural networks. It was shown that the application of the classical experimental design theory makes it possible to obtain a neural network with considerably smaller uncertainty than the one designed without it. The author hopes that the material presented in this paper will encourage engineers to use the discussed techniques in industrial practice.

Acknowledgments

The author would like to express his sincere gratitude to the referees whose efforts have significantly improved the quality of the paper. The work was supported by the Ministry of Education and Science in Poland under the grant 4T11A01425 *Modelling and identification of non-linear dynamic systems in robust diagnostics*.

References

Alessandri A., Parisini T. and Zoppoli R (1997): *Neural approximators for non-linear finite-memory state estimation*. — Int. J. Contr., Vol. 67, No. 2, pp. 275–302.

Atkinson A.C. and Donev A.N. (1992): *Optimum Experimental Designs*. — New York: Oxford University Press.

Bäck T., Hammel U. and Schwefel H.P. (1997): *Evolutionary computation: Comments on the history and current state*. — IEEE Trans. Evolut. Computat., Vol. 1, No. 1, pp. 3–17

Blanke M., Kinnaert M., Lunze J. and Staroswiecki M. (2003): *Diagnosis and Fault-Tollerant Control*. — Berlin: Springer

Chandra P. and Sing Y. (2004): *Feedforward sigmoidal networks – Equicontinuity and fault tolerance*. — IEEE Trans. Neural Netw., Vol. 15, No. 6, pp. 1350–1366.

Chen J. and Patton R. J. (1999): *Robust Model-based Fault Diagnosis for Dynamic Systems*. — London: Kluwer.

Chen J., Patton R. J. and Liu G.P. (1996): *Optimal residual design for fault diagnosis using multi-objective optimization and genetic algorithms*. — Int. J. Syst. Sci., Vol. 27, No. 6, pp. 567–576.

Chen Z., He Y., Chu F. and Huang J. (2003): *Evolutionary strategy for classification problems and its application in fault diagnosis*. — Eng. Appl. Artif. Intell., Vol. 16, No. 1, pp. 31–38.

Cohn D.A. (1994): *Neural network exploration using optimal experimental design*, In: Advances in Neural Information Processing Systems 6 (J. Cowan *et al.*, Eds.). — San Mateo: Morgan Kaufman.

Chryssolouris G., Lee M. and Ramsey A. (1996): *Confidence interval prediction for neural network models*. — IEEE Trans. Neural Netw., Vol. 7, No. 1, pp. 229–232.

DAMADICS (2004): Website of *DAMADICS*: *Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems*. `http://diag.mchtr.pw.edu.pl/damadics/`.

Delebecque F., Nikoukah R. and Rubio Scola H. (2003): *Test signal design for failure detection: A linear programming approach*. — Int. J. Appl. Math. Comput. Sci., Vol. 13, No. 4, pp. 515–526.

Fedorov V.V., and Hackl P. (1997): *Model-Oriented Design of Experiments*. — New York: Springer.

Fleming P.J. and Purshouse (2002): *Evolutionary algorithms in control systems engineering: A survey*. — Contr. Eng. Pract., Vol. 10, No. 11, pp. 1223–1241.

Fogel D.B. (1995): *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. — New York: Willey-IEEE Press.

Fogel L.J., Owens A.J. and Walsh M.J. (1999): *An overview of evolutionary programming*, In: Evolutionary Algorithms (De Jong K., Vose M.D. and Whitley L.D., Eds.). — Heidelberg: Springer.

Frank P.M. and Köppen-Seliger (1997): *New developments using AI in fault diagnosis*. — Eng. Appl. Artif. Intell., Vol. 10, No. 1, pp. 3–14.

Frank P.M., Schreier G. and Garcia E.A. (1999): *Non-linear observers for fault detection and isolation*, In: New Directions in Non-linear Observer Design (Nijmeijer H., Fossen T.I., Eds.). — Berlin: Springer.

Fuente M.J. and Saludes S. (2000): *Fault detection and isolation in a non-linear plant via neural networks*. — Proc. Symp. *Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS*, Budapest, Hungary, pp. 472–477.

Fukumizu K. (1996): *Active learning in multilayer perceptrons*, In: Advances in Neural Information Processing Systems (D.S. Touretzky *et al.*, Eds.). — Cambridge: MIT Press, pp. 295–301.

Fukumizu K. (2000): *Statistical active learning in multilayer perceptrons*. — IEEE Trans. Neural Netw., Vol. 11, No. 1, pp. 17–26.

Galar R. (1989): *Evolutionary search with soft selection*. — Biol. Cybern., Vol. 60, No. 1, pp. 124–141.

Goodwin G.C. and Payne R.L. (1977): *Dynamic Systems Identification. Experiment Design and Data Analysis*. — New York: Acadamic Presss.

Gray G. J, Murray-Smith D. J., Li Y., Sharman K. C. and Weinbrenner T. (1998): *Non-linear model structure identification using genetic programming*. — Contr. Eng. Pract., Vol. 6, No. 11, pp. 1341–1352.

Guo L.Z. and Zhu Q.M. (2002): *A fast convergent extended Kalman observer for non-linear discrete-time systems*. — Int. J. Syst. Sci., Vol. 33, No. 13, pp. 1051–1058.

Gupta M.M., Jin L. and Homma N. (2003): *Static and Dynamic Neural Networks. From Fundamentals to Advanced Theory*. — Hoboken, NJ: Wiley.

Holland J.H. (1975): *Adaptation in Natural and Artificial Systems*. — Ann Arbor, MI: The University of Michigan Press.

Hunt K.J., Sbarbaro D., Zbikowski R. and Gawthrop P.J. (1992): *Neural networks for control systems – A survey*. — Automatica, Vol. 28, No. 6, pp. 1083–1112.

Ivakhnenko A.G. and Mueller J.A. (1995): *Self-organizing of nets of active neurons*. — Syst. Anal. Modell. Simul., Vol. 20, pp. 93–106.

Janczak A. (2005): *Identification of Non-Linear Systems Using Neural Networks and Polynomial Models: A Block-Oriented Approach*. — Berlin: Springer.

Karpenko M., Sepehri N. and Scuse D. (2003): *Diagnosis of process valve actuator faults using a multilayer neural network*. — Contr. Eng. Practice, Vol. 11, No. 11, pp. 1289–1299.

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

Kowalczuk Z., Suchomski P. and Białaszewski T. (1999): *Evolutionary muti-objective Pareto optimization of diagnostic state observers*. — Int. J. Appl. Math. Comput. Sci., Vol. 9, No. 3, pp. 689–709.

Koza J. R. (1992): *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. — Cambridge: The MIT Press.

Le T.T., Watton J. and Pham D.T. (1997): *An artificial neural network based approach to fault diagnosis and classification of fluid power systems*. — J. Syst. Contr. Eng., Vol. 211, No. 4, pp. 307–317.

Le T.T., Watton J. and Pham D.T. (1998): *Fault classification of fluid power systems using a dynamics feature extraction technique and neural networks*. — J. Syst. Contr. Eng., Vol. 212, No. 2, pp. 87–97.

MacKay D. (1992): *Information-based objective functions for active data selection*. — Neural Comput., Vol. 4, No. 4, pp. 305–318.

Marcu T. (1997): *A multiobjective evolutionary approach to pattern recognition for robust diagnosis of process faults*. — Proc. Symp. *Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS*, Hull, UK, pp. 1183–1188.

Metenidis M.F., Witczak M. and Korbicz J. (2004): *A novel genetic programming approach to non-linear system modelling: Application to the DAMADICS benchmark problem*. — Eng. Appl. Artif. Intell., Vol. 17, No. 4, pp. 363–370.

Michalewicz Z. (1996): *Genetic Algorithms + Data Structures = Evolution Programs*. — Berlin: Springer.

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

Miller J.A., Potter W.D., Gandham R.V. and Lapena C.N. (1993): *An evaluation of local improvement operators for genetic algorithms*. — IEEE Trans. Syst. Man and Cybern., Vol. 23, No. 5, pp. 1340–1351.

Mrugalski M. and Witczak M. (2002): *Parameter estimation of dynamic GMDH neural networks with the bounded-error technique*. — J. Appl. Comput. Sci., Vol. 10, No. 1, pp. 77–90.

Narendra K.S. and Parthasarathy K. (1990): *Identification and control of dynamical systems using neural networks*. — IEEE Trans. Neural Netw., Vol. 1, No. 1, pp. 1–27.

Patan K. and Parisini T. (2005): *Identification of neural dynamic models for fault detection and isolation: the case of a real sugar evaporation process*. — J. Proc. Contr., Vol. 15, No. 1, pp. 67–79.

Porter L.L. and Passino K.M. (1995): *Genetic adaptive observers*. — Eng. Appl. Artif. Intell., Vol. 8, No. 3, pp. 261–269.

Ruano A.E. (Ed.) (2005): *Intelligent Control Systems Using Computational Intelligence Techniques*. — London: The IEE Press.

Sjoberg J., Zhang Q., Ljung L., Benveniste A., Delyon B., Glorennec P. Y., Hjalmarsson H. and Juditsky A. (1995): *Nonlinear black-box modeling in system identification: A unified overview*. — Automatica, Vol. 31, No. 12, pp. 1691–1724.

Sun R., Tsung F. and Qu L. (2004): *Combining bootstrap and genetic programming for feature discovery in diesel engine diagnosis*. — Int. J. Ind. Eng., Vol. 11, No. 3, pp. 273–281.

Uciński D. (2005): *Optimal Measurements Methods for Distributed Parameter System Identification*. — Boca Raton, FL: CRC Press.

Walter E. and Pronzato L. (1997): *Identification of Parametric Models from Experimental Data*. — Berlin: Springer

Watton J. and Pham D.T. (1997): *An artificial NN based approach to fault diagnosis and classification of fluid power systems*. — J. Syst. Contr. Eng., Vol. 211, No. 4, pp. 307–317.

Weerasinghe M., Gomm J.B and Williams D. (1998): *Neural networks for fault diagnosis of a nuclear fuel processing plant at different operating points*. — Contr. Eng. Pract., Vol. 6, No. 2, pp. 281–289.

Witczak M., Obuchowicz A. and Korbicz J. (2002): *Genetic programming based approaches to identification and fault diagnosis of non-linear dynamic systems*. — Int. J. Contr., Vol. 75, No. 13, pp. 1012–1031.

Witczak M. (2003): *Identification and Fault Detection of Nonlinear Dynamic Systems*. — Zielona Góra: University of Zielona Góra Press.

Witczak M. and Korbicz J. (2004): *Observers and genetic programming in the identification and fault diagnosis of non-linear dynamic systems*, In: Fault Diagnosis. Models, Artificial Intelligence, Applications (Korbicz J., Kościelny J.M., Kowalczuk Z. and Cholewa W., Eds.). — Berlin: Springer.

Witczak M., Korbicz J., Mrugalski M. and Patton R.J. (2006): *GMDH neural network-based approach to robust fault detection and its application to solve the DAMADICS benchmark problem*. — Contr. Eng. Pract., Vol. 14, No. 6, pp. 671–683.

Witczak M. and Prętki P. (2005): *Designing neural-network-based fault detection systems with D-optimum experimental conditions*. — Comput. Assist. Mech. Engi. Sci., Vol. 12, No. 2, pp. 279–291.

Witczak M. (2006): *Towards the training of feed-forward neural networks with the D-optimum input sequence*. — IEEE Trans. Neural Netw., Vol. 17, No. 2, pp. 357–373.

White H. (1989): *Learning in artificial neural networks: A statistical perspective*. — Neural Comput., Vol. 4, No. 4, pp. 305–318.

Yen G.G. and Lin K. (2000): *Wavelet packet feature extraction for vibration monitoring*. — IEEE Trans. Ind. Electron., Vol. 47, No. 3, pp. 650-667.

Zhang J., Ma J. and Yan Y. (2000): *Assessing blockage of the sensing line in a differential-pressure flow sensor by using the wavelet transform of its output*. — Meas. Sci. Technol., Vol. 11, No. 3, pp. 178–184.

Zhao J., Chen B. and Shen J. (1998): *Multidimensional nonorthogonal wavelet-sigmoid basis function neural networkfor dynamic process fault diagnosis*. — Comput. Chem. Eng., Vol. 23, pp. 83–92.

Zolghardi A., Henry D. and Monision M. (1996): *Design of nonlinear observers for fault diagnosis. A case study.* — Contr. Eng. Pract., Vol. 4, No. 11, pp. 1535–1544.

Received: 12 November 2005
Revised: 2 February 2006