Abstract:
The paper deals with the problems of robust fault detection using analytical methods (observers and unknown input observers) and soft computing techniques (neural networks, neuro-fuzzy networks and genetic programming). The model-based approach to Fault Detection and Isolation (FDI) is considered. In particular, observers for non-linear Lipschitz systems and extended unknown input observers are discussed. In the case of soft computing techniques, the main objective is to show how to employ the bounded-error approach to determine the uncertainty of the GMDH and neuro-fuzzy networks. It is shown that based on soft computing models uncertainty defined as a confidence range for the model output, adaptive thresholds can be defined. The final part of the paper presents two illustrative examples that confirm the effectiveness of the unknown input observers and the neuro-fuzzy networks approaches.

Keywords: Fault detection, unknown input observer, dynamical neural networks, neuro-fuzzy systems, evolutionary algorithms

1. Introduction
A continuous increase in the complexity, efficiency, and reliability of modern industrial systems necessitates a continuous development in the control and fault diagnosis theory and practice (Blanke et al., 2003; Chen and Patton, 1999; Isermann, 2006; Korbicz et al., 2004; Kościelny, 2001; Rodrigues et al., 2007). These requirements extend beyond the normally accepted safety-critical systems of nuclear reactors, chemical plants or aircrafts, to new systems such as autonomous vehicles or fast rail systems. An early detection and maintenance of faults can help avoid system shutdown, breakdowns and even catastrophes involving human fatalities and material damage.

The core of the fault detection and isolation system is the so-called model-based approach. In the general case, this concept can be implemented using various kinds of models: analytical, knowledge-based and data-based ones (Köppen-Seliger and Frank, 1999), which are used to model a diagnosed system working in normal-operation or faulty conditions. Conventional model-based fault detection techniques make use of analytical or quantitative models (Korbicz et al., 2004), mostly in the framework of observers or Kalman filters (Chen and Patton, 1999; Witzczak, 2007; Zolghadri et al., 1996). The dynamic behaviour of the system is described by differential equations or transfer functions together with the respective parameter values.

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. Currently many efforts are made to use knowledge-based or qualitative or data-based models (Patton and Korbicz, 1999; Korbicz, 2006a). They represent system behaviour in terms of heuristic or qualitative knowledge (Frank, 1990; Korbicz, 2004). The relationship between inputs and outputs may be described by a rule base or by a set of parameters that have to be determined during an identification stage based on the learning data set. In this case data-based models, such as neural networks (Korbicz et al., 1999), fuzzy sets (Frank and Köppen-Seliger, 1997; Kowal, 2005), the evolutionary algorithms (Chen and Patton, 1999; Obuchowicz, 2003; Witczak et al., 2002) or their combination (Patton et al., 2005; Korbicz et al., 2001), 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 the model used to represent system behaviour, the better the chance of improving the reliability and performance in diagnosing faults. Indeed, disturbances as well as model uncertainty are inevitable in industrial systems (Patton et al., 2006), and hence there exists a pressure creating the need for robustness in fault diagnosis systems. 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. In one way or another, all the above-mentioned approaches can realise this requirement for linear systems.

Taking into account the above conditions, a large amount of knowledge on designing robust fault diagnosis systems has been accumulated through the literature since the beginning of the 1980s. For a comprehensive survey regarding such techniques, the reader is referred to the excellent monographs (Chen and Patton, 1999; Gertler, 1998; Korbicz et al., 2004; 2002; Patton et al., 2000). The most common approach to robust fault diagnosis is to use robust observers. This is mainly because of the fact that the theory of robust observers is relatively well developed in the control engineering literature.

The main objective of this survey paper is to present recent developments in modern fault diagnosis with non-linear observers, neural networks, neuro-fuzzy networks...
and genetic programming. In particular, the paper is organised as follows: Section 2 outlines the problem of observer-based robust fault diagnosis and presents two different observer structures that can be employed for non-linear systems. The genetic programming approach in modelling of diagnosed systems in Section 3 is presented. Section 4 presents alternative 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 structure optimization problem when the mathematical state-space model is not available. The neuro-fuzzy structure optimization problem when the mathematical state-space model is not available.

### 2. Analytical models in fault detection systems

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 system is most important part. In general analytical models applied to fault detection can be divided in following groups (Chen and Pattan, 1999; Gertler, 1998; Korbicz et al., 2004):

- physical models (equations of movement, balance equations, etc.),
- input-output type linear models (continuous or discrete transmittances),
- state linear and non-linear equations.

Among these models a special role belongs to state space equations that are applied for designing of state observers (Korbicz and Bidyuk, 1993).

The basic idea underlying observer-based (or filter-based, in the stochastic case) approaches to fault detection is to obtain the estimates of certain measured and/or unmeasured signals. Then, in the most usual case, the estimates of the measured signals are compared with their originals, i.e., the difference between the original signal and its estimate is used to form a residual signal $r_c = y_k - \hat{y}_k$. To tackle this problem, many different observers (or filters) can be employed, e.g., Luenberger observers, Kalman filters, etc.

The admiration for observer-based fault detection schemes is caused by the still increasing popularity of state-space models as well as the wide usage of observers in modern control theory and applications. Due to such conditions, the theory of observers (or filters) seems be well developed (especially for linear systems). This has made a good background for the development of observer-based FDI schemes.

Irrespective of the linear or non-linear FDI technique being employed (Witczak, 2003), FDI performance will be usually impaired by the lack of robustness to model uncertainty. As can be observed in the literature (Chen and Pattan, 1999; Gertler, 1998; Korbicz et al., 2004; Patton et al., 2000), the most common approach to robust fault diagnosis is to use robust observers. In particular, the so-called unknown input model uncertainty is mostly preferred. The observer resulting from such an approach is called the Unknown Input Observer (UIO). Although the origins of UIOs can be traced back to the early 1970s (cf. the seminal work of Wang et al. (1975)), the problem of designing such observers is still of paramount importance both from the theoretical and practical viewpoints (Hui and Z̆ak, 2005). The main objective of the subsequent part of this section is to present two unknown input observer design strategies that can be employed for the Lipschitz (Korbicz et al., 2007; Witczak and Korbicz, 2006; Witczak et al., 2006b) and a general class of non-linear systems (Witczak et al., 2002; 2006), respectively.

#### 2.1. Observers for non-linear Lipschitz systems

Let us consider Lipschitz systems that can be described as follows:

$$
x_{k+1} = Ax_k + Bu_k + h(y_k, u_k) + g(x_k, u_k),
$$

$$
y_{k+1} = Cx_{k+1},
$$

where $x_k \in \mathbb{R}^n$ stands for the state vector, $y_k \in \mathbb{R}^m$ is the output, $u_k \in \mathbb{R}$ is the input, and $g(\cdot)$ and $h(\cdot)$ are non-linear functions. Additionally, $g(\cdot)$ satisfies

$$
\|g(x_1, u) - g(x_2, u)\| \leq \gamma \|x_1 - x_2\|, \quad \forall x_1, x_2, u,
$$

and $\gamma > 0$ stands for the Lipschitz constant.

Let us consider an observer for the system (1)-(2) described by the following equation:

$$
\hat{x}_{k+1} = A\hat{x}_k + Bu_k + h(y_k, u_k) + g(\hat{x}_k, u_k) + K(y_k - C\hat{x}_k),
$$

where $\hat{x}_k$ denotes the state estimate and $K$ stands for the gain matrix.

The subsequent part of this section shows theorem that presents convergence conditions of (4). Following Thau (1973), let us assume that the pair $(A, C)$ is observable. Let $P = P^T$, $P > 0$ be a solution of the following Lyapunov equation:

$$
Q - P = A^T P A_0 - A - A^T K C,
$$

where $A_0$ is a stable matrix, i.e., $\rho(A_0) < 1$, and $Q = Q^T$, $Q > 0$. Moreover, let $\gamma(\cdot)$ and $\sigma(\cdot)$ stand for the minimum and maximum singular values of its argument, respectively.

**Theorem 1.** (Witczak and Korbicz, 2006) Let us consider the observer (4) for the systems described by (1)-(2). If the Lipschitz constant $\gamma$ (cf. (3)) satisfies

$$
\gamma < \sqrt{\frac{2(Q - A^2_0 P A_0)}{\sigma(P)^2 + 1}}, \quad Q - A_0^2 P A_0 > 0,
$$

then the observer (4) is asymptotically convergent.

Unfortunately, the condition (6) may merely serve as a method of checking the convergence, but the gain matrix $K$ has to be determined beforehand. This means that the design procedure boils down to selecting various gain
matrices $K$, solving the Lyapunov equation (5), and then checking the convergence condition (6). There is no doubt that this is an ineffective and inconvenient approach.

To tackle such a challenging problem, an effective design procedure was proposed in (Witczak and Korbicz, 2006), which can be written as follows:

**Step 1:** Obtain $\gamma$ for (1)-(2).

**Step 2:** Solve a set of linear matrix inequalities:

\[
\begin{bmatrix}
\beta & P \\
P & \beta
\end{bmatrix} \succ 0, \quad \beta > 0, \quad P > 0,
\]

where $\beta$ is a given positive constant. \hspace{1cm} (7)

\[
X \succ 0,
\]

\[
\begin{bmatrix}
X & A^T P - C^T L^T \\
P A - LC & \beta
\end{bmatrix} \succ 0,
\]

where $P$ is a symmetric matrix. \hspace{1cm} (8)

\[
\begin{bmatrix}
P \beta & X \\
PA - LC & \beta
\end{bmatrix} \succ 0.
\]

**Step 3:** Obtain the gain matrix $K = P^{-1}L$.

In the next part of this section a straightforward approach for extending the techniques proposed in the preceding sections to discrete-time Lipschitz systems with unknown inputs is presented. The system can be described as follows:

\[
x_{k+1} = Ax_k + Bu_k + h(y_k, u_k) + g(x_k, u_k) + Ed_k,
\]

\[
y_{k+1} = Cx_{k+1},
\]

where $d_k \in \mathbb{R}^l$ is the unknown input, and $E$ is a known unknown input distribution matrix. In order to use the techniques described above for state estimation of the system (10)-(11), it is necessary to introduce some modifications concerning the unknown input. Let us assume that

\[
\text{rank}(CE) = \text{rank}(E) = q.
\]

(see (Chen and Patton, 1999, p. 72, Lemma 3.1). If the condition (12) is satisfied, then it is possible to calculate $H = (CE)^{-1} = (CE^T)^{-1}CE^T$, where $^T$ stands for the pseudo-inverse of its argument. By multiplying (11) by $H$ and then inserting (10), it can be shown that

\[
x_{k+1} = \bar{A} x_k + \bar{B} u_k + \bar{h}(y_k, u_k) + \bar{g}(x_k, u_k) + \bar{E} y_{k+1},
\]

\[
\bar{A} = CA, \quad \bar{B} = CB, \quad \bar{g}(\cdot) = CG(\cdot), \quad \bar{h}(\cdot) = CH(\cdot), \quad \bar{E} = EH.
\]

Thus, the unknown input observer for (10)-(11) is given as follows:

\[
\hat{x}_{k+1} = \hat{A} x_k + \hat{B} u_k + \hat{h}(y_k, u_k) + \hat{g}(x_k, u_k) + \hat{E} y_{k+1} + K(y_k - C\hat{x}_k).
\]

A simple comparison of (1) and (13) leads to the conclusion that the observer (14) can be designed with the above-mentioned three-step procedure, taking into account the fact that:

\[
\|g(x_1, u) - g(x_2, u)\| \leq \gamma \|x_1 - x_2\|, \quad \forall x_1, x_2, u,
\]

and assuming that the pair $(A, C)$ is observable.

### 2.2. Extended unknown input observers and genetic programming

Let us consider a non-linear discrete-time system described by

\[
x_{k+1} = g(x_k) + h(u_k) + E_k d_k,
\]

\[
y_{k+1} = C x_{k+1},
\]

where $d_k$ is the unknown input, and $E_k$ is the known matrix. Using the similar approach as in Section 2.1, it can be shown (Witczak et al., 2006) that the structure of the so-called Extended Unknown Input Observer (EUIO) is

\[
\hat{x}_{k+1} = \hat{x}_{k+1}/k - K_{k+1}(y_{k+1} - C_{k+1}\hat{x}_{k+1}/k),
\]

where

\[
\hat{x}_{k+1}/k = g(\hat{x}_k) + h(\hat{u}_k) + E_k y_{k+1}.
\]

As a consequence, the algorithm used for state estimation of (16)-(17) can be given as follows:

\[
\hat{x}_{k+1}/k = g(\hat{x}_k) + h(\hat{u}_k) + E_k y_{k+1},
\]

\[
P_{k+1}/k = A_k P_k A_k^T + Q_k,
\]

\[
K_{k+1} = P_{k+1}/k (C_{k+1} P_{k+1}/k C_{k+1}^T + R_k)^{-1},
\]

\[
\hat{x}_{k+1} = \hat{x}_{k+1}/k + K_{k+1}(y_{k+1} - C_{k+1}\hat{x}_{k+1}/k),
\]

\[
P_{k+1} = (I - K_{k+1} C_{k+1}) P_{k+1}/k.
\]

where

\[
A_k = \frac{\partial g(x_k)}{\partial x_k} \bigg|_{x_k = \hat{x}_{k-1}} - C_k \frac{\partial g(x_k)}{\partial x_k} \bigg|_{x_k = \hat{x}_{k-1}} - C_k A_k.
\]

Employing the Lyapunov approach to convergence analysis of the EUIO it can be proved that the domain of attraction significantly depends on the covariance matrices $Q_{k-1}$ and $Q_{k-1}$ of the process $w_k$ and measurement $v_k$ noise, respectively. Unfortunately, an analytical derivation of the $Q_{k-1}$ and $R_k$ matrices seems to be an extremely difficult problem. However, it is possible to set the above matrices as $Q_{k-1} = \beta_1 I$, $R_k = \beta_2 I$, with $\beta_1$ and $\beta_2$ large enough. On the other hand, it is well-known that the convergence rate of such an Extended Kalman Filter (EKF)-like approach can be increased by an appropriate selection of the covariance matrices $Q_{k-1}$ and $R_k$, i.e., the more accurate (near “true” values) the covariance matrices, the better the convergence rate. This means that in the deterministic case ($w_k = 0$ and $v_k = 0$) both matrices
should be zero. Unfortunately, such an approach usually leads to the divergence of the observer as well as to other computational problems. To tackle these difficulties, a compromise between the convergence and the convergence rate should be established.

Using (23), the state estimation error (for fault-free mode) can be given as

\[ e_{k+1} = x_{k+1} - \hat{x}_{k+1} = \left( I - K_{k+1}C_{k+1} \right) e_{k+1/\hat{u}} \tag{26} \]

and

\[ e_{k+1/\hat{u}} = x_{k+1} - \hat{x}_{k+1/\hat{u}} = \mathbf{g}(x_k) - \mathbf{g}(\hat{x}_k) - \alpha_k \hat{A}_k e_k \tag{27} \]

where \( \alpha_k = \text{diag}(\alpha_{k,1}, \ldots, \alpha_{k,n}) \) is an unknown diagonal matrix. Thus, using (27), the equation (26) becomes

\[ e_{k+1} = \left( I - K_{k+1}C_{k+1} \right) \alpha_k \hat{A}_k e_k \tag{28} \]

It is clear from (27) that \( \alpha_k \) represents the linearisation error. First, let us define

\[ \sigma_k = \max_{j=1,\ldots,n} |\alpha_{k,j}|, \quad \sigma_k = \min_{j=1,\ldots,n} |\alpha_{k,j}|. \tag{29} \]

**Theorem 2.** (Witczak et al., 2006) If

\[ \sigma_k \leq \left( \frac{\mathcal{L}(A_1) \mathcal{L}(C_{11})^2 + (A_1, P_1 A_1^T + Q_1)}{\sigma (C_{11} P_{11} C_{11}^T + R_{11})} \right)^{1/2}, \tag{30} \]

where \( 0 < \epsilon < 1 \), then the proposed extended unknown input observer is locally asymptotically convergent.

It is clear from (30) that the bound of \( \sigma_k \) can be maximised by suitable settings of the instrumental matrices \( Q_k \) and \( R_k \). This can be realised as follows (Witczak et al., 2006):

\[ Q_k = (\gamma \epsilon_k^2 + \delta_1) I, \quad R_k = \delta_2 I \tag{31} \]

with \( \gamma > 0 \) and \( \delta_1 > 0, \delta_2 > 0 \) are large and small enough, respectively. On the other hand the instrumental matrices can be set as follows:

\[ Q_{k+1} = \delta_3 I \tag{32} \]

\[ R_k = \gamma e_k^T I \tag{33} \]

where \( \delta_1, \delta_2, \delta_3 \) are nonlinear functions of the output error \( \epsilon_k \) (the squares are used to ensure the positive definiteness of \( Q_k \) and \( R_k \)). Thus, the problem reduces to identifying the above functions. To tackle this problem, Genetic Programming (GP) (Obuchowicz, 2003; Koza, 1992) can be employed. The unknown functions \( \psi(e_{k+1}) \) and \( \tau(e_k) \) can be expressed as a tree. Thus, in the case of \( \psi(\cdot) \) and \( \tau(\cdot) \), the terminal sets are \( T - (e_{k+1}) \) and \( T - (e_k) \), respectively. In both cases, the function set can be defined as \( \mathcal{F} = \{ +, *, /, \psi(\cdot), \ldots, \psi(\cdot) \} \), where \( \psi(\cdot) \) is a nonlinear univariate function and, consequently the number of populations is \( n_p = 2 \). Since the terminal and function sets are given, the GP approach can be easily adapted for the identification purpose of \( \psi(\cdot) \) and \( \tau(\cdot) \). First, let us define the performance index including a necessary ingredient of the \( Q_k \) and \( R_k \) selection process.

Since the instrumental matrices should be chosen in order to maximize the convergence rate, we have

\[ (Q_k, R_k) = \arg \max_{(Q_k, R_k)} J_{\Delta k}(\psi(e_k), \tau(e_k)) \tag{35} \]

where \( J_{\Delta k}(\psi(e_k), \tau(e_k)) = \sum_{k=0}^{n-1} \text{trace} P_k \).

On the other hand, owing to FDI requirements, it is clear that the output error should be closed to zero in the fault-free mode. In this case, one can define another performance index:

\[ (Q_k, R_k) = \arg \min_{(Q_k, R_k)} J_{\Delta k}(\psi(e_k), \tau(e_k)), \tag{36} \]

where

\[ J_{\Delta k}(\psi(e_k), \tau(e_k)) = \sum_{k=0}^{n-1} \epsilon_k^T e_k. \tag{37} \]

Therefore, in order to couple (35) and (36), the following identification criterion is employed:

\[ (Q_k, R_k) = \arg \min_{(Q_k, R_k)} J_{\Delta k}(\psi(e_k), \tau(e_k)) \tag{38} \]

Since the identification criterion is established, it is straightforward to use the GP algorithm.

The numerical example considered here is a fifth-order two-phase nonlinear model of an induction motor (Boutayeb and Aubry, 1999). Moreover, the following three cases concerning the selection of \( Q_k \) and \( R_k \) were considered:

**Case 1:** Classical approach (constant values), i.e., \( Q_0 = 0.1, R_0 = 0.1 \).

**Case 2:** Selection supported by an analytical consideration:

\[ Q_k = 10^4 \epsilon_k^T I, R_k = 0.01 I. \tag{39} \]

**Case 3:** GP-based approach. In order to obtain the matrices \( Q_k \) and \( R_k \) using the GP-based approach (Case 3) (Witczak and Korbić, 2004), a set of \( n_t = 300 \) input-output measurements was generated. The simulation results (for all the cases) are shown in Fig. 1 (Witczak et al., 2002). It can be seen, that the proposed approach is superior to the classical technique of selecting the instrumental matrices \( Q_k \) and \( R_k \).
3. Genetic programming in modelling

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, real-world 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 (Goldberg, 1989; Michalewicz, 1996; Obuchowicz, 2003). Genetic Programming (GP) (Koza, 1992; Obuchowicz, 2003) is an extension of EAs. The main difference between these two approaches is that in GP the evolving individuals are parse trees rather that fixed-length binary strings (Kowalczuk and Białaszewski, 2006).

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 (Nelles, 2001; Walter and Pronzato, 1997; Janczak, 2005). Knowing that the diagnosed system exhibits nonlinear characteristics, a choice of the nonlinear model set must be made. In this section, an NARX (Nonlinear AutoRegressive 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:

\[
y_k = g_k(y_{k-1}, \ldots, y_{k-n_{1,y}}, \ldots, \hat{y}_{m,k-1}, \ldots, \hat{y}_{m,k-n_{1,y}}, u_{i,k-1}, \ldots, u_{i,k-n_{1,y}}, u_{r,k-1}, \ldots, u_{r,k-n_{1,y}}, \theta_i),
\]

where \(g_k = g_k(y_{k-1}, u_{k-1}, \ldots, u_{k-n_{1,y}}, \theta_i)\) and \(\theta_i\) is the model parameters vector. Thus the system output is given by

\[
y_k = \hat{y}_k + e_k,
\]

where \(e_k\) consists of a structural deterministic error, caused by the model-reality mismatch, and the measurement noise \(\sigma_k\). The problem is to determine an unknown function \(g() = (g_{1,k}, \ldots, g_{m,k})\) 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 \(39\) as a tree, or a set of trees. Indeed, as is shown in Fig. 2, the Multi-Input and Single-Output (MISO) NARX model can be easily put in the form of a tree, and hence to build the MIMO model \(39\) it is necessary to use \(m\) trees. In such a tree, two sets can be distinguished, namely, the terminal \(T\) and function \(F\) sets. 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.

![Fig. 2. Exemplary GP tree representing the model \(y_k = \hat{y}_{k-1} + u_{k-1} + \hat{y}_{k-2}/u_{k-2}\)](image)

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. In (Esparcia-Alcazar, 1998), a tree representation is 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) (Walter and Pronzato, 1997), where the following quality index is minimized:

\[ J_{AIC}(M_i) = \frac{1}{2} J(M_i(\hat{\theta})) + \frac{1}{n_T} \dim \theta', \]

(41)

where \( J(M_i(\hat{\theta})) = \ln \det \sum_{k=1}^{n_T} \varepsilon_k \varepsilon_k^T \),

(42)

and \( \hat{\theta} = \arg \min_{\theta} J(M_i(\theta)) \) 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) (DAMADICS, 2002). Figure 3 illustrates the obtained results (Witczak et al., 2002).

3.2. Choice of the gain matrix for the robust nonlinear observer

The solution of the diagnosed system modeling presented in the previous subsection possesses a disadvantage. Usually, the parameters of the obtained GP model have no physical interpretations. Here a nonlinear state observer designing methodology based on the classical approach and a GP technique and proposed by Witczak and co-workers (2002) is presented. Consider the nonlinear discrete system

\[ x_{k+1} = f(x_k, u_k, \nu_k), \]
\[ y_k = h(x_k, v_k), \]

(43)

where \( u_k \) is the input, \( x_k \) is the output, \( y_k \) is the state, \( \nu_k \) and \( v_k \) represent the process and measurement noise, and \( f(\cdot), h(\cdot) \) are nonlinear functions.

The problem is to estimate the state \( x_k \) of the system (43), where a set of measured inputs and outputs and the model of the system are given. The classical methods using different kinds of approximation are often applied (Korbicz and Bidyuk, 1993), and can be given as follows:

\[ \hat{x}_k = \hat{x}_k^0 + K_k \varepsilon_k, \]
\[ y_k = h(x_k, v_k), \]

(44)

(45)

where \( \epsilon_k \) denotes the \( a \ priori \) output error, \( \hat{x}_k \) is an \( a \ priori \) state estimate, \( \hat{x}_k \) is a state estimate and \( K_k \) is the gain matrix.

The gain matrix \( K_k \) of the observer (44)-(45) can be searched for by various methods (e.g., the Kalman filter, the Luenberger observer, etc.) which, in a large majority, consist of constant elements and are not robust to model uncertainties. In (Witczak et al., 1999), the gain matrix is composed of certain functions, i.e., each entry of the gain matrix is a function which depends on the \( a \ priori \) output error and the system input. Therefore, it can be written as follows:

\[ \hat{x}_k = \hat{x}_k^0 + K(\epsilon_k, u_k)\epsilon_k. \]

(46)

Thus the main goal is to obtain an appropriate form of \( K(\epsilon_k, u_k) \) based on a set of measured outputs and inputs and the mathematical model of the system. Even if the mathematical model is uncertain and/or the initial state is far from its expected value, it seems possible to obtain \( K(\epsilon_k, u_k) \) to ensure the best fitness to the real system. For that purpose, a GP technique is exploited, where the gain matrix is obtained off-line from a randomly created population by means of an evolutionary process.

3.3. GP approach to the state-space representation of the system

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

\[ x_{k+1} = g(x_k, u_k) + \nu_k, \]
\[ y_{k+1} = C x_{k+1} + \nu_k. \]

(47)

(48)

Assume that the function \( g(\cdot) \) has the form

\[ g(x_k, u_k) = A(x_k) x_k + h(u_k). \]

(49)

Thus, the state-space model of the system (47) can be expressed as

\[ \hat{x}_{k+1} = A(\hat{x}_k) \hat{x}_k + h(u_k), \]
\[ y_{k+1} = C(\hat{x}_{k+1} + \nu_k). \]

(50)

(51)

Without loss of generality, it is possible to assume that

\[ A(\hat{x}_k) = \text{diag}(a_{ik}(\hat{x}_k)) \quad | i = 1, 2, \ldots, n). \]

(52)

The problem reduces to identifying the nonlinear functions \( a_{ik}(\hat{x}_k), h_i(u_k) (i = 1, \ldots, n) \), and the matrix \( C \). Assuming \( \max_{k=1,\ldots,n} | a_{ik}(\hat{x}_k) | < 1 \), it can be shown (Witczak et al., 2002) that the model (50) is globally asymptotically...
stable. This implies that $a_{ri}(\hat{x}_i)$ should have the following structure:

$$a_{ri}(\hat{x}_i) = \tanh \left( s_{ri}(\hat{x}_i) \right), \quad i = 1, \ldots, n,$$

(53)

where $\tanh(\cdot)$ is a hyperbolic tangent function, and $s_{ri}(\hat{x}_i)$ is a function to be determined.

In order to identify $h_i(\hat{x}_i), h_i(u_i) \ (i = 1, \ldots, n)$ and the matrix $C$, the GP algorithm is applied. The fitness function is defined by (41)-(42).

The GP algorithm was successfully applied to build a model of the apparatus at the Lublin Sugar Factor S.A. (Poland) (DAMADICS, 2002). Figure 4 illustrates obtained results (Witczak, 2002).

For the model-based approach (Korbicz et al., 2004; 2002; Patton et al., 2000), the neural network replaces the analytical model that describes the process under the normal operating conditions (Frank and Köppen-Seliger, 1997; Korbicz, 2006; Patton et al., 2005). 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) (Gupta et al., 2003; Osowski, 2006; Tadeusiewicz, 1993).

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 (Gupta et al., 2003; Norgard et al., 2000; Patan, 2007), 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 self-organizing maps (Haykin, 1999; Korbicz et al., 1994).

As was mentioned in Section 2, when non-linear state space models are available, fault diagnosis can be realised by using the concept of an unknown input observer. 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. If the above-mentioned approach fails, then describing model uncertainty in a different way seems to be a good remedy. One of the 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 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.

The 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, bounds the residual at a level that is dependent on model uncertainty, and hence it provides a more reliable fault detection.

Contrary to the typical industrial applications of neural networks that are presented in the literature (Karpenko et al., 2003; Korbicz et al., 2004; Mrugalski and Korbicz, 2006), Witczak et al. (2006a) defined the task of designing a neural network in such a way as to obtain a model with a 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 a relatively large uncertainty. A degraded performance of fault diagnosis constitutes a direct consequence of using such models.

To tackle this challenging problem for non-linear dynamic systems, the GMDH (Group Method of Data Handling) approach (Ivakhnenko and Mueller, 1995; Korbicz and Mrugalski, 2007) can be effectively adapted (Witczak et al., 2006a). A complete design procedure concerning the application of GMDH neural networks to robust fault detection is proposed. 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, they obtained a tool that is able to generate an adaptive threshold. 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 a 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.

As has been mentioned, 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 minimise model uncertainty as far as possible. This can be realised with the application of Optimum Experimental Design (OED) theory (Atkinson and Donev, 1992; Uciński, 2005; Walter and Pronzato, 1997). Recently, Witczak (2006b) developed a D-optimum experimental design strategy that can be used for training single-output neural networks.

Robust GMDH neural networks. A successful application of the ANNs to the system identification and fault diagnosis tasks (Korbicz et al., 2001) 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 (Ivakhnenko and Mueller, 1995; Mrugalski, 2004; Mrugalski and Korbicz, 2005) 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 (Korbicz and Kuś, 1999; Patan and Parisini, 2005). 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 (Mrugalski, 2004; Witczak et al., 2006a).

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.

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 (Mrugalski, 2004) that the application of the classical evaluation criteria such as the Akaike Information Criterion (AIC) and the Final Prediction Error (FPE) (Ivakhnenko and Mueller, 1995; Mueller and Lemke, 2000) can lead to the selection of inappropriate neurons and, consequently, to unnecessary structural errors.

Apart from the model structure selection stage, inaccuracy in parameter estimates also contributes to modelling uncertainty. Indeed, while applying the least-square method to parameter estimation of neurons, a set of restrictive assumptions has to be satisfied (Witczak et al., 2006a). An effective remedy to such a challenging problem is to use the so-called Bounded Error Approach (BEA) (Milanese et al., 1996; Witczak et al., 2006a). Let us consider the following system:

\[ y_k = r_k^T \theta + \varepsilon_k. \]  

where \( r_k \) stands the regressor vector, \( \theta \in \mathbb{R}^n \) denotes the parameter vector, and \( \varepsilon_k \) 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

\[ \bar{y}_k^m \leq y_k \leq \bar{y}_k^M, \]  

where \( \bar{y}_k^m \) and \( \bar{y}_k^M \) 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 $\varepsilon_k$ consists of a structural
deterministic error caused by the model-reality mismatch,
and the stochastic error caused by the measurement noise
is bounded as follows:
\begin{equation}
\varepsilon^m_k \leq \varepsilon_k \leq \varepsilon^M_k,
\end{equation}
where the bounds $\varepsilon_k$ and $\varepsilon^M_k$ ($\varepsilon^m_k \neq \varepsilon^M_k$) can be estimated
(Witczak et al., 2006a).

The idea underlying the bounded-error approach is to
obtain a feasible parameter set $\mathbb{P}$ (Milanese et al., 1996)
that is consistent with the input-output measurements
used for parameter estimation. The resulting $\mathbb{P}$ is described
by a polytope defined by a set of vertices $\mathbb{V}$. Thus, the
problem of determining the model output uncertainty can
be solved as follows:
\begin{equation}
r^T \theta^m \leq r^T \theta \leq r^T \theta^M,
\end{equation}
where
\begin{equation}
\theta^m = \arg \min_{\theta \in \mathbb{P}} r^T \theta, \quad \theta^M = \arg \max_{\theta \in \mathbb{V}} r^T \theta.
\end{equation}

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 above-presented
approach for the uncertain regressor case, let us denote an
unknown "true" value of the regressor $r_{nk}$ by a difference
between the measured value of the regressor $r_k$ and the
error in the regressor $e_k$:
\begin{equation}
r_{nk} = r_k - e_k,
\end{equation}
where it is assumed that the error $e_k$ is bounded as
\begin{equation}
e^m_k \leq e_k \leq e^M_k, \quad i = 1, \ldots, n_p.
\end{equation}

Using (54) and substituting (59) into (60), one can
define the space containing the parameter estimates:
\begin{equation}
\varepsilon^m_k - \varepsilon^m \theta \leq y_k - r^T \theta \leq \varepsilon^M + e^M_k,
\end{equation}
which makes it possible to adapt the above-described
technique to the error-in-regressor case (Witczak et al.,
2006a).

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:
\begin{equation}
y^m_k \leq r^T \theta \leq y^M_k.
\end{equation}

In order to adapt the presented approach to parameter
estimation of non-linear neurons with an activation function $\ell(\cdot)$, it is necessary to transform the relation
\begin{equation}
\varepsilon^m_k \leq \ell(r^T \theta) \leq \varepsilon^M_k,
\end{equation}
using $\ell^{-1}(\cdot)$ and hence
\begin{equation}
\xi^{-1} (y_k - \varepsilon^m_k) \leq r^T \theta \leq \xi^{-1} (y_k - \varepsilon^M_k).
\end{equation}

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. 5).

![Fig. 5. Illustration of the concept of the adaptive threshold](image)

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
\begin{equation}
y^m_k \leq y_k \leq y^M_k + e^M_k.
\end{equation}

This means that robust fault detection boils down to
checking if the output of the system satisfies (65). Thus,
when (65) is violated, then a fault symptom occurs.

5. Neuro-fuzzy networks in fault detection
The procedure of Neuro-Fuzzy (NF) network design
consists of the structure selection stage and the parameter
estimation stage (Korbicz and Kowal, 2001; Rutkowski,
2002; Rutkowski, 2005; Piegat, 2003). 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
(Babuška, 1998; Kowal and Korbicz, 2002a; 2002b; 2003;
Uppal et al., 2006). 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 neuro-fuzzy 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.
Two main strategies for placing fuzzy sets in the input space can be distinguished: the first one proposes to minimize the output error of the global model (Leith and Leithead, 1999), and the other one is based on partial models that model the local behavior of the system (Abonyi et al., 2002). A typical property of the first approach is to arrange fuzzy sets in the input space in such a way that all partial models are active in the whole domain of input variables. In this case, the accuracy of the global model is guaranteed by the proper mixture of partial model outputs. The alternative approach does not examine the global accuracy of the model but concentrates on partial models, which should tune in to the local behavior of the system. The problem of rule base declaration reduces to the determination of the number of rules required for a precise description of the problem to be solved.

The simplest method used to determine the number of rules is based on generating a uniformly distributed grid of rules in the input space. The usage of such an approach is limited to simple systems with a small number of inputs. The approach does not work well for more complicated systems because it generates a combinatorial explosion of rules, which make this method useless. Fuzzy clustering algorithms are another technique which is often used for fuzzy rule generation (Babuška, 1998; Chen et al., 1998; Kowal et al., 2002; Mendes et al., 2002). The idea of this approach is to find natural groups of data in order to apply to each group one fuzzy rule. It seems to be natural to use fuzzy clustering algorithms in the case of neuro-fuzzy networks. The task of fuzzy clustering is usually reduced to finding the local minimum of the nonlinear cost function, defined by the following expression:

\[ J(X, U, V) = \sum_{i=1}^{N} \sum_{k=1}^{c} u_{ik}^m D_{ik}^2, \]

(66)

where the matrix \( U \) contains the membership degrees of data points from the matrix to the defined clusters \( X, V = [v_1, v_2, ..., v_c], v_i \in R^n \) is a matrix which defines the centers of the clusters, \( D_{ik} \) is a metric used to determine the distance between the data points and the cluster centers:

\[ D_{ik}^2 = \|x_k - v_i\|^2 = (x_k - v_i)^T A (x_k - v_i), \]

(67)

and the parameter \( m \) takes values from 1 to \( oo \) and determines the degree of fuzziness of the clusters. The cost function (66) can be viewed as a total variance of the data \( x_k \) with respect to the cluster centers \( v_i \). The matrix \( A \) which occurs in the expression (67) is used to tune the shape and orientation of the clusters in the space.

The fuzzy clustering algorithm which uses such a norm to calculate the distance between data points and cluster centers is called Fuzzy C-Mean (FCM). However, the number of the found clusters strongly depends on the values of coefficients, which must be defined by the designer at the beginning of the procedure, so the application of the algorithm is difficult. Two clustering algorithms were applied to build the model of the valve which is a part of the industrial installation of the Lublin Sugar Factory (DAMADICS, 2002). The learning procedure of the Takagi-Sugeno neuro-fuzzy network was divided into two phases. In the first step, clustering methods were used to optimize the network structure and prepare the initial values of the parameters. In the second step, the gradient descent method was used to tune all parameters. Two clustering algorithm were used in the first step: the mountain method and the fuzzy C-mean algorithm (Kowal et al., 2002; Mendes et al., 2002). Sample results are shown in Fig. 6.

Fig. 6. Performance of the TSK neuro-fuzzy model for the valve V (SSE / N.° of samples = 6.774)

5.1. Robust neuro-fuzzy networks

The application of neuro-fuzzy networks in diagnostic areas (Calado et al., 2006) 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 (Chen and Patton, 1999; Mrugalski, 2003; Witczak, 2003). 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 Bounded Error Approach (BEA) method 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 LP system (Kowal, 2005; Kowal and Korbicz, 2005a). 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 the proposed approach the clustering algorithm is used to determine the ellipsoid clusters in the input-output space in order to generate for each found cluster one local linear submodel and to determine the parameters of the fuzzy partitions by cluster projection (Babuška, 1998). Another approach is based on the detection of approximately linear dependencies in the data space using a modified BEA (Kowal, 2005; Kowal and Korbicz, 2005a; 2005b). The algorithm consists in the generation of a single rule for each found linear dependency and allows the parameters of fuzzy partitions.

In order to present the BEA approach for estimating the parameters of the determining dynamic Takagi-Sugeno (T-S) network, let us consider the following T-S neuro-fuzzy model:

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

(68)

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

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

(69)

The model described by the equation (68) can be viewed in the form of an LP system:

$$y = x_k^T \theta,$$

(70)

if the parameters of the fuzzy sets are treated like constant values. Here $x_k$ denotes the input vector containing the delayed inputs $y_i$ of the local models and the delayed output $y_k$ of the local models, i.e.:

$$x_k = [y_k, y_k^{k-1}, \ldots, y_k^{k-n_k}, y_k^{k-1}, y_k^{k-2}, \ldots, y_k^{k-n_k}]$$

The output error is given by the following formulae:

$$e_k = y_k - x_k^T \theta,$$

(71)

where $e_k$ is the error and $y_k$ is the output of the system. The error is bounded by means of the following inequalities:

$$e_k^\min \leq e_k \leq e_k^\max.$$

(72)

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

$$P = \{\theta \in \mathbb{R}^n | y_k^\min - e_k^\max \leq x_k^T \theta \leq y_k^\max - e_k^\min, \ k = 1, \ldots, N\}.$$  

(73)

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. 7. If the maximum and minimum values of the parameters are known,

$$\theta_i^\min = \arg \min_{\theta \in P} \theta_i,$$

(74)

$$\theta_i^\max = \arg \max_{\theta \in P} \theta_i,$$

(75)

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

$$\theta_i = \frac{\theta_i^\min + \theta_i^\max}{2}, \ i = 1, \ldots, N.$$  

(76)

The minimum and maximum values for the following parameters are determined using the linear programming technique (Milanese et al., 1996). The feasible set of parameters is used also to compute the confidence interval for the output of the system:

$$x_k^T \theta_i^\min + e_k^\min \leq y_k \leq x_k^T \theta_i^\max + e_k^\max,$$

(77)

where

$$\theta_i^\max = \arg \max_{\theta \in \mathbb{R}^n} x_k^T \theta,$$

(78)

$$\theta_i^\min = \arg \min_{\theta \in \mathbb{R}^n} x_k^T \theta.$$  

(79)
The confidence interval can be directly applied to calculate the adaptive threshold for the residual signal:

\[ r_k = y_k' - y_k. \]  

(80)

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

\[ x_k^T \theta_k^{\min} + \epsilon_k^{\min} - y_k \leq x_k^T \theta_k^{max} + \epsilon_k^{max} - y_k. \]  

(81)

Unfortunately, the computations required to determine all vertices \( W \) of 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 (Milanese et al., 1996). One of the proposed solution is the Outer Bounding Ellipsoid method which has been applied to fault detection in a DC engine (Kowal, 2005; Kowal and Korbicz, 2006).

6. Illustrative examples

The main objective of this section is to present two examples which illustrate the effectiveness of the approaches described in the preceding sections. In particular, the first example is devoted to robust fault diagnosis of an induction motor with the extended unknown input observer. The second one concerns the fault detection of an intelligent actuator with the Takagi-Sugeno network.

6.1. Observer-based fault detection of an induction motor

The numerical example considered here is a fifth-order two-phase non-linear model of an induction motor, which has already been the subject of a number of various control design applications (Boutayeb and Aubry, 1999). The complete discrete-time model in a stator-fixed \((a,b)\) reference frame is

\[
\begin{align*}
x_{1,k+1} &= x_{1,k} + h \left( -\gamma x_{1,k} + \frac{K}{L_c} x_{3,k} + K_p x_{4,k} x_{4,k} + \frac{1}{\sigma L_c} \right), \\
x_{2,k+1} &= x_{2,k} + h \left( -\gamma x_{2,k} - K_p x_{4,k} x_{4,k} + \frac{1}{\sigma L_c} \right), \\
x_{3,k+1} &= x_{3,k} + h \left( \frac{M}{L_c} x_{1,k} - \frac{1}{\sigma L_c} x_{3,k} - p x_{5,k} x_{5,k} \right), \\
x_{4,k+1} &= x_{4,k} + h \left( \frac{M}{L_c} x_{2,k} + p x_{5,k} x_{5,k} - \frac{1}{\sigma L_c} x_{4,k} \right), \\
x_{5,k+1} &= x_{5,k} + h \left( \frac{M}{L_c} x_{3,k} x_{5,k} - x_{4,k} \right) - \frac{r}{J} T_c \\
y_{1,k+1} &= x_{1,k+1}, \quad y_{2,k+1} = x_{2,k+1}, \\
\end{align*}
\]  

(82)

(83)

(84)

(85)

(86)

(87)

where \( x_k = [x_{1,k}, \ldots, x_{6,k}]^T = [i_{s1k}, \psi_{s1k}, \psi_{s2k}, \psi_{s3k}, \omega_k]^T \) represents the currents, the rotor fluxes, and the angular speed, respectively, while \( u_k = [\delta_{i1k}, \delta_{i2k}]^T \) is the stator voltage control vector, \( p \) is the number of the pairs of poles, and \( \tau_c \) is the load torque. The rotor time constant \( T_r \) and the remaining parameters are defined as

\[ T_r = \frac{L_c}{p}, \quad \sigma = 1 - \frac{M^2}{L_c L_r}, \quad K = \frac{M}{\sigma L_c}, \quad \gamma = \frac{K}{\sigma L_c}, \quad R_s \frac{M^2}{\sigma L_c L_e}. \]  

(88)

where \( R_s, R_r \) and \( L_s, L_r \) are stator and rotor per phase resistances and inductances, respectively, and \( J \) is the rotor moment inertia.

The numerical values of the above parameters are as follows: \( R_s = 0.18 \, \Omega, \quad R_r = 0.15 \, \Omega, \quad M = 0.068 \, H, \quad L_s = 0.0699 \, H, \quad L_r = 0.0699 \, H, \quad J = 0.0558 \, k g m^2, \quad T_L = 10 \, \text{Nm}, \quad p = 1, \quad a \) and \( h = 0.1 \, \text{ms}. \) The input signals are

\[ u_{1,k} = 350 \cos(0.03k), \quad u_{2,k} = 300 \sin(0.03k). \]  

(89)

The initial conditions for the system and the observer are \( x_k = 0 \) and \( \hat{x}_k = [200, 200, 50, 50, 50]^T, \) and \( P_0 = 10^6 I, \)

\[ Q_{k-1} = 10^6 \epsilon_{\epsilon_{k-1}} \epsilon_{u_{k-1}} I = 0.001 I, \]

\[ R_k = 0.01 I. \]  

(90)

Let us assume that the unknown input distribution matrix is

\[ E = [1.2, 0.2, 2.4, 1, -1.6]^T, \]  

(91)

and the corresponding unknown input is simulated by

\[ d_k = 3.0 \sin(0.5 \pi k) \cos(0.03 \pi k). \]  

(92)

Thus, the system (16)(17) is described using (82)-(87) and 88. The following fault scenarios were considered (Witczak, 2007):

Case 1: Abrupt fault of the \( y_{1,k} \) sensor:

\[ f_{1,k} = \begin{cases} 0, & 500 < k < 1400, \\ -0.3 y_{1,k}, & \text{otherwise}, \end{cases} \]  

(93)

and \( f_{2,k} = 0. \)

Case 2: Abrupt fault of the \( u_{1,k} \) actuator:

\[ f_{1,k} = \begin{cases} 0, & 500 < k < 1400, \\ -0.3 y_{1,k}, & \text{otherwise}, \end{cases} \]  

(94)

and \( f_{2,k} = 0. \)

Thus, the system is now described by

\[ x_{1,k+1} = g(x_k) + h(u_k) + E_k d_k + L_{1,k} f_k, \]

\[ y_{1,k+1} = C_{k-1} x_{1,k+1} + L_{2,k-1} f_{k-1}, \]  

(95)

(96)

with (82)-(87), 88, \( f_k = [f_{1,k}, f_{2,k}]^T, \) and

\[ L_{1,k} = \begin{bmatrix} \frac{1}{r} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad L_{2,k} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \]  

(97)

From Figs. 8 and 9, it can be observed that the residual signal is sensitive to the faults under consideration, which confirms its reliability and abilities of unknown input decoupling.
This, together with unknown input decoupling, implies that the process of fault detection becomes a relatively easy task.

Fig. 8. Residuals for a sensor fault

Fig. 9. Residuals for an actuator fault

6.2. Neuro-fuzzy-based fault detection of an intelligent actuator

The scheme of the actuator with an intelligent positioner is given in Fig. 10. Such actuator has been investigated by international research group during the realization of the so-called DAMADICS (2002) project. In the Fig. 10 the following notations are used: V1, V2 and V3 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 denoted is placement, pressure and volume flow transducers, respectively. For remote on-line diagnostics, the following measured variables are accessible: the flow rate of juice after the control valve (F), the actuator’s rod displacement (X), the input set-point (CV), juice temperature at the input of the control valve (T1), and juice pressures at the input and outlet of the control valve, respectively (P1 and P2).

Applying the method for structure generation of NF models (Kowal, 2005; Kowal and Korbicz, 2005b) and the results presented in Sub-section 5.1, two NF models can be defined. The obtained structures are described in Table 1.
The parameters 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, additionally, their system output uncertainty. Figure 11 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) occurred.

From Fig. 12, 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 (DAMADICS, 2002) was tested as well.

From the point of view 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.

Unfortunately, most of such systems exhibit non-linear behaviour, which makes it impossible to use the well-developed techniques for linear systems. If it is assumed that the system is linear, which is not true in general, and even if robust techniques for linear systems are used (e.g., unknown input observers), it is clear that such an approximation may lead to unreliable fault detection and, consequently, an early indication of faults which are developing is rather impossible. Such a situation increases the probability of the occurrence of faults, which can be extremely serious in terms of economic losses, environmental impact, or even human mortality. Indeed, robust techniques are able to tolerate a certain degree of model uncertainty. In other words, they are not robust to everything, i.e., are robust to an arbitrary degree of model uncertainty. This real world development pressure creates the need for new techniques which are able to tackle fault diagnosis of non-linear systems. In spite of the fact that the problem has been attacked from various angles by many authors and a number of relevant results have already been reported in the literature, there is no general framework which can be simply and conveniently applied to maintain fault diagnosis for non-linear systems.

Taking into account the above discussion, the main objective of this paper was to consider a robust model-based fault detection system applying analytical and soft computing models. Special attention was paid to the uncertainty of such models 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 TakagiSugeno networks can be used to develop an adaptive threshold that permits robust fault detection. In the last part, an experimental study performed with the non-linear model of an induction motor and the DAMADICS benchmark problem showed the effectiveness of such robust fault detection based on the
extended unknown input observer and the uncertainty of neuro-fuzzy models.

<table>
<thead>
<tr>
<th>No</th>
<th>Description</th>
<th>S</th>
<th>M</th>
<th>B</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Control valve faults</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f1</td>
<td>Valve clogging</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>f2</td>
<td>Sedimentation</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>f3</td>
<td>Seat erosion</td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>f4</td>
<td>Bushing frictions</td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>f5</td>
<td>External leakage</td>
<td></td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>f6</td>
<td>Internal leakage</td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>f7</td>
<td>Medium evaporation</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Servo-motor faults</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f8</td>
<td>Twisted piston rod</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>f9</td>
<td>Housing</td>
<td></td>
<td></td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>f10</td>
<td>Diaphragm perforation</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>f11</td>
<td>Spring fault</td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>Positioner faults</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f12</td>
<td>E/P transducer fault</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>f13</td>
<td>Rod displ. sensor fault</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>f14</td>
<td>Pressure sensor fault</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>f15</td>
<td>Feedback fault</td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>External faults</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f16</td>
<td>Pressure drop</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>f17</td>
<td>Unexpected pressure change</td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>f18</td>
<td>Opened bypass valves</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>f19</td>
<td>Flow rate sensor fault</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>

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

AUTHOR
Józef Korbicz – Deputy Rector for Scientific Research and International Cooperation and professor at the Institute of Control and Computation Engineering, University of Zielona Góra, Poland, e-mail: J.Korbicz@issi.uz.zgora.pl.

References
At: http://diag.mchtr.pw.edu.pl/damadics


via GMDH neural networks, Proc. 16-th IFAC World Congress, Prague, Czech Republic, CD-ROM.


