Fault diagnosis of dynamical systems using evolutionary algorithms

Marcin Witczak [1] and Józef Korbicz [1]

[1] University of Zielona Góra, Institute of Control and Computation Engineering, Zielona Góra, Poland, email: {M.Witczak,J.Korbicz}@issi.uz.zgora.pl

Abstract. Challenging and complex design problems arise regularly in modern fault diagnosis systems. Unfortunately, the classical analytical techniques cannot often provide acceptable solutions to such difficult tasks. This explains why soft computing techniques such as evolutionary algorithms are becoming 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 in fault diagnosis. The main attention is on the techniques that integrate the classical and evolutionary approaches. A selected example, dealing with the DAMADICS benchmark, is carefully described in the paper.

1 Introduction

A continuous increase in the complexity, efficiency, and reliability of modern industrial systems necessities a continuous development in the control and fault diagnosis [12] theory and practice. These requirements extend beyond normally accepted safety-critical systems of nuclear reactors, chemical plants or aircraft, to new systems such as autonomous vehicles or fast rail systems. An early detection and maintenance of faults can help avoid the system shut-down, breakdown and even catastrophes involving human fatalities and material damage. The role of the fault diagnosis system 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 the three subtasks: fault detection, fault isolation, and fault identification [5]. The 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 required system performance.

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 [5, 12]. There is also a number of different approaches that can be employed to settle the robustness problems regarding model uncertainty [5, 12]. 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 [5, 12], they all usually suffer from the lack of an appropriate mathematical description of the system being considered. If there is no, or not sufficiently accurate analytical models, then the one feasible way is to use the so-called soft computing techniques [12]. 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., [5, 9, 12, 20, 24] and the references therein).

The paper is organised as follows: Section I is devoted to a bibliographical review of the application of evolutionary algorithms in fault diagnosis. Sections III and IV present the genetic programming-based techniques that can be used for designing observers and state-space models for non-linear systems. The presented approaches are illustrated with the results concerning the DAMADICS benchmark [4]. The final section summarises and concludes the paper.

2 Evolutionary algorithms in FDI

Evolutionary algorithms have been the subject of a large number of papers [1]. Generally, there is a large number of different kinds of EAs and the most popular of them are: genetic algorithms (GAs) [11], genetic programming (GP) [14], evolutionary programming [8], evolutionary strategies [17], and evolutionary search with soft selection [10].

Although the origins of evolutionary algorithms can be traced back to the late 1950's (see [1] for a comprehensive introduction and survey on EAs), the first works on evolutionary algorithms in control engineering were published at the beginning of the 1990's. In 2002, Fleming and Purshouse [7] tackled a challenging task of preparing a comprehensive survey on the application of evolutionary algorithms in control engineering. As indicated in [7], there are relatively few publications on applications of evolutionary algorithms to the design of FDI systems.

In this paper, rather than providing an exhaustive survey on evolutionary algorithms in fault diagnosis it 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. In other words, the main objective is to extend the material of [7] 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 [5, 12]. Undoubtedly, the most common approach is to use robust observers, such as the Unknown Input Observer (UIO) [5, 12, 24], 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 is 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 impair the fault sensitivity considerably. In order to settle this problem, Chen *et al.* [2] (see also [5]) formulated an 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.* [13], where the observer design is based 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 for 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 [12]. Unfortunately, this strategy works well only when the linearization does not cause a large mismatch between the linear model and non-linear behavior. 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 [26]. To overcome this problem, Porter and Passino proposed the so-called genetic adaptive observer [19]. 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 the convergence conditions of the observer. They also did not consider the robustness issues with respect to model uncertainty. A solution that does not posses such drawbacks was proposed by Witczak *et al.* [23]. 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 three- (two- or even four-) tank system, an inverted pendulum, a travelling crane, etc. To tackle this problem the genetic programming-based approach for designing state-space models from input-output data was developed in [23, 24]. This approach will be detailed in Section 2.1 A further development of this technique related with the input-output models was proposed in [16]. Evolutionary algorithms have also been applied to FDI methods that are not based on the concept of residuals. Marcu [15] formulated the 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 symptoms [18]. The method relied upon the availability of a priori probabilities that a particular fault caused a particular symptom. In [3], the authors employed genetic algorithm-based evolutionary strategy for fault diagnosis-related classification problems, which includes two aspects: evolutionary selection of the training samples and input features, and evolutionary construction of the neural network classifier. Finally, Sun *et al.* [21] used the bootstrap technique to preprocess the operational data acquired from a running diesel engine and the genetic programming approach to find the best compound feature that can discriminate among 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:

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

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

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, $\boldsymbol{g}\left(\cdot\right)$ 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 the 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 estimate, 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

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

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 obtain $\boldsymbol{A}(\cdot)$ and $\boldsymbol{h}(\cdot)$. Assuming that $|a_{i,i}(\hat{\boldsymbol{x}}_k)| < 1, i = 1, \ldots, n$ it can be shown [23] that the model (4)-(5) is globally asymptotically stable [23]. 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 a 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) [14]. 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 non-linear (in general) functions they can easily be represented as trees. The language of the trees in GP is formed by a user-defined function $\mathbb{F}$ set and terminal $\mathbb{T}$ set, which form the nodes of the trees. In the case of a parameterized tree, the terminal set is composed of variables only. Such a parameterization has proven to be especially useful for a model designing purpose [23, 24, 25]. 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 [22].

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 the two terminal sets can be distinguished, i.e. first for $\boldsymbol{A}(\hat{\boldsymbol{x}}_k)$ ($\mathbb{T} = \{\hat{\boldsymbol{x}}_k\}$) and second for $\boldsymbol{h}(\boldsymbol{u}_k)$ ($\mathbb{T} = \{\boldsymbol{u}_k\}$).

As can be observed, parameter estimation involves 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-square method, it is possible to obtain $\boldsymbol{C}_k$ (assuming $\boldsymbol{C}_k = \boldsymbol{C}$) by solving the following 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 be also pointed out that the order n of the model is in general unknown and hence should be determined throughout experiments.

2.2 Observer design with genetic programming

Let us consider a class of non-linear system described by the following equations

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

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

where $\boldsymbol{g}\left(\boldsymbol{x}_k\right)$ 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 [12], the UIO presented in [5, 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):

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

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

and

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

where the way of calculating the gain $\boldsymbol{K}_{1,k+1}$ and unknown input decoupling $\boldsymbol{H}_{k+1}$ matrices is given in [23, 24, 25]. The main objective of is to show that the convergence of the EUIO strongly depends on the appropriate choice of the instrumental matrices $\boldsymbol{R}_k$ and $\boldsymbol{Q}_k$ (measurement and process noise covariance matrices in the stochastic case). Thus, 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.
For that purpose Witczak *et al.* (2002) performed a comprehensive convergence analysis with the Lyapunov method. Unfortunately, an analytical derivation of the matrices $\boldsymbol{Q}_{k-1}$ and $\boldsymbol{R}_k$ is an extremely difficult problem. To tackle this, a compromise between the convergence and the convergence rate should be established. As shown in [23, 24, 25], such a compromise can be attained as follows:

$$\boldsymbol{Q}_{k-1} = q^2(\boldsymbol{\varepsilon}_{k-1})\boldsymbol{I} + \delta_1\boldsymbol{I}, \quad \boldsymbol{R}_k = r^2(\boldsymbol{\varepsilon}_k)\boldsymbol{I} + \delta_2\boldsymbol{I}, \tag{14}$$

where $q(\boldsymbol{\varepsilon}_{k-1})$ and $r(\boldsymbol{\varepsilon}_k)$ are non-linear functions of the output error $\boldsymbol{\varepsilon}_k$ (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 [23] it is shown how to reduce the task of designing of $\boldsymbol{Q}_{k-1}$ and $\boldsymbol{R}_k$ to the multi-objective and global structure optimization problem of $q(\boldsymbol{\varepsilon}_{k-1})$ and $r(\boldsymbol{\varepsilon}_k)$. In particular, genetic programming is utilized [23, 24, 25] to tackle this problem.

2.3 An illustrative example – DAMADICS benchmark

The DAMADICS (*Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems*) [4] project was focused on the diagnosis of valve (cf. Fig. 1) plant actuators and looked towards real implementation methods for new actuator systems. As can be seen in Fig. 1, the following process variables can be measured:

$\boldsymbol{u}_k = (CV, P1, P2, T1)$ and $\boldsymbol{y}_k = (F, X)$ where 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. 1, 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 objective of this section is to design the state-space model of the actuator being considered (cf. Fig. 1) according to the approach described in Section 2.1. The terminal set used during the identification process was $\mathbb{F} = \{+, *, /\}$. After the 50 runs of the GP algorithm performed for each model order, $n = 2, \ldots, 8$, 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 [25]).

Since the state-space model is given, then 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 [25] was employed, then the approach of Section III along with the threshold selector described in [25] were used for fault detection. Since the method

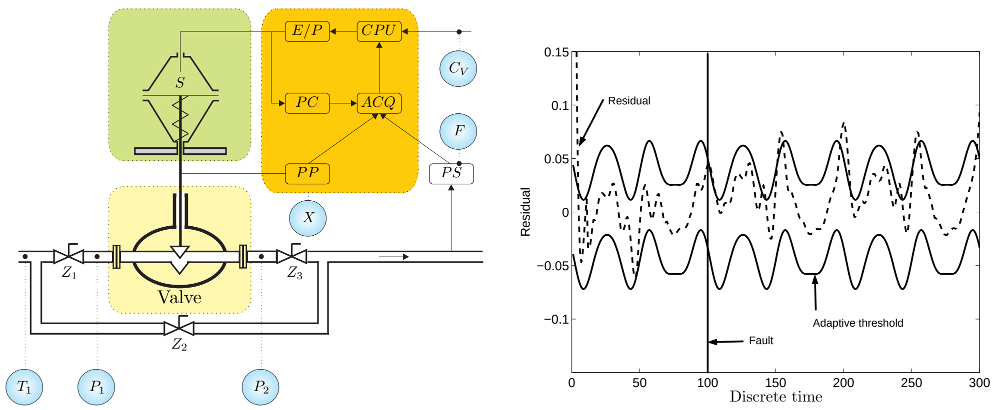

Figure 1. Actuator scheme (left) and the residual $r_{1,k}$ for the flow sensor fault.

of designing an appropriate threshold is known, it is possible to check the fault detection capabilities of the presented observer-based fault detection scheme. Fig. 1 presents results of fault detection for 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, $r_{1,k} = y_{1,k} - \hat{y}_{1,k}$. A complete report regarding fault diagnosis of the DAMADICS benchmark can be found in [25].

3 Conclusions

The main objective of this paper was to summarize the advances in the interesting research area concerning application of evolutionary algorithms to fault diagnosis. Rather than providing an exhaustive survey it was aimed at providing a comprehensive account of the published work that exploits the special nature of these techniques. Special attention was paid to the techniques that integrate the classical and soft computing approaches in such a way to tackle the inevitable non-linearity and robustness problems. In particular, it was shown that the integration of genetic programming and unknown input observer results in a hybrid fault diagnosis technique that is superior over the classical one. The presented technique was illustrated with the DAMADICS benchmark.

Acknowledgments

The work was supported by Ministry of Scientific Research and Information Technology in Poland under the grant 4T11A01425

Bibliography

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

[2] 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.

[3] Chen Z., He Y., Chu F., and Huang J. (2003): *Evolutionary strategy for classification problems andits application in fault diagnosis.* – Engineering Applications of Artificial Intelligence, Vol. 16, No. 1, pp. 31–38.

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

[5] Chen J., and Patton R. J. (1999): *Robust Model-based Fault Diagnosis for Dynamic Systems.* — London: Kluwer Academic Publishers

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

[7] Fleming P.J., and Purshouse (2002): *Evolutionary algorithms in control systems engineering: a survey.* – Control Eng. Practice, Vol. 10, No. 11, pp. 1223–1241.

[8] 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-Verlag).

[9] Frank P.M., and Köppen-Seliger (1997): *New developments using AI in fault diagnosis.* – Engineering Applications of Artificial Intelligence, Vol. 10, No. 1, 3–14.

[10] Galar, R. (1989): *Evolutionary search with soft selection.* — Bio. Cyber., Vol. 51, pp. 124-141.

[11] Holland, J.H. (1975): *Adaptation in natural and artificial systems.* — Ann Arbor, MI: The University of Michigan Press.

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

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

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

[15] 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.

[16] 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.* – Engineering Applications of Artificial Intelligence, Vol. 17, No. 4, 363–370.

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

[18] 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. Sys., Man and Cyber., Vol. 23, No. 5, pp. 1340–1351.

[19] Porter L.L. and Passino K.M. (1995): *Genetic adaptive observers.* – Engineering Applications of Artificial Intelligence, Vol. 8, No. 3, pp. 261–269.

[20] Ruano A.E. (Ed.) (2005): *Intelligent Control Systems using Computational Intelligence Techniques.* – London: The IEE Press.

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

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

[23] 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.

[24] Witczak M. (2003): *Identification and Fault Detection of Non-linear Dynamic Systems.* — Zielona Góra: University of Zielona Góra Press

[25] Witczak M. and Korbicz J. (2004): *Observers and genetic programming in the indentification 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-Verlag.

[26] Zolghardi A., Henry D. and Monision M. (1996): *Design of non-linear observers for fault diagnosis. A case study.* — Control. Eng. Practice. Vol. 4, No. 11, pp. 1535-1544.