Neuro-Fuzzy Learning and Genetic Algorithm Approach with Chaos Theory Principles Applying for Diagnostic Problem Solving

Stefania GALLOVA, *Member, IAENG*

Abstract— **Performance results for finding the best genetic algorithm for the complex real problem of optimal machinery equipment operation and predictive maintenance are presented. A genetic algorithm is a stochastic computational model that seeks the optimal solution to an objective function. A methodology calculation is based on the idea of measuring the increase of fitness and fitness quality evaluation with chaos theory principles applying within genetic algorithm environment. Fuzzy neural networks principles are effectively applied in solved manufacturing problems mostly where multisensor integration, real - timeness, robustness and learning abilities are needed. A modified Mamdani neuro-fuzzy system improves the interpretability of used domain knowledge.**

Index Terms— **chaos theory, fuzzy rule, fitness, genetic algorithm, Mamdani neuro-fuzzy system, metric entropy.**

I. Introduction

The complexity and reliability demands of contemporary industrial systems and technological processes require the development of new fault diagnosis approaches. Due to the large number of process variables and their complex interconnections in a machinery environment, the pertinent knowledge system is mostly qualitative and incomplete.

The diagnostic parameters of machine can be represented by a linguistic variable with fuzzy set definition through "fuzzification" in terms of a defined fuzzy membership function. Fuzzy rules describe the qualitative relations between the major operation conditions (i.e. various operation conditions, vibrations, tool accuracy, safe load, working load and so on). In practice, classical tools solve this task by using of sensors great number, or by special sensors creating. This approach increases costs and reduces the reliability. It is possible to compensate this unfavourable state effectively by higher quality of analysis at lower requirements for definiteness of inputs [1].

Chaos theory is a progressive field, which in this time is not sufficiently utilized at failure analysis as possible effective tool. Nowadays, used tools of analysis, or statistic parameters of random phenomena do not find out all possibilities, that is all possible information about real technical state of equipments that the given random signal could provide. Chaos theory principles provide very valuable complementary information, which is not possible to obtain by another means of analysis. Genetic algorithm is a suitable medium for effective application of chaos theory principles for real diagnostic problem solving. It is possible to work with inaccurate, vague, indefinite, non-unambiguous information that real practice in predominant extent provides.

Fuzzy neural networks principles are effectively applied in solved manufacturing problems as a first step mostly where multisensor integration, real - timeness, robustness and learning abilities are needed. A modified Mamdani neuro-fuzzy system improves the interpretability of used domain knowledge by parameter data processing.

These progressive approaches to complex diagnostic problem solving of practice provide the high effective tools for optimal machine condition and for security of machine equipment reliability.

Manuscript received March 23, 2009. This work was supported by the Slovak Republic's Ministry of Education under Scientific Grant Project:
Intelligent Approach to Automated Diagnostic Problem Solving of Machinery Equipment (Number: VEGA 1/4133/07).
Stefania GALLOVA. Author is with the Technical University of Kosice, Letna 9, SK-042 00 Kosice, Slovak Republic; phone: +421-904-584-426; e-mail: stefania.gallova@zoznam.sk; stefania.gallova@tuke.sk

II. A Mamdani Neuro-Fuzzy Network Module

Since the higher levels of the control and the monitoring hierarchy require symbolic representation of knowledge and processing techniques, the integrated use of the symbolic and subsymbolic approaches is straightforward. Neuro-fuzzy technique implementation contains the following tasks. Determination of membership functions by self-organized clustering, selection of relevant fuzzy rules by competitive learning, elimination and combination of rules and adjustment of the parameters of the membership functions by supervised backpropagation learning are realized [2] - [6]. The neural models can be easily applied to the residual generation and to perform the first stage of the fault diagnosis that relies on the fault detection. Unfortunately, subsequent stage of the fault diagnosis called fault identification demands determination of the type of a fault, its size and cause, which may be difficult to obtain with the application of the neural model-based approaches. In order to solve these problems the approach, which relies on the parameters identification of diagnosed system, can be applied. The idea of the proposed methodology is based on the parameters estimation of the diagnosed system. The comparison of the estimated parameters with the a priori estimated parameters of the nominal system allows performing both the fault detection and identification.

Neural networks applications contain three phases to solving any problem. At first, we have training during which weights of network connections are changed. Output of network is compared to the training data. Error of solved neural network is evaluated. The network is verified within the second phase. Values of connections weights are constant and the solved neural network is checked if its output is the same as in the training phase. Generalization procedure is the last phase when the network output is evaluated for such data, which were not used for training procedure the network.

Neuro-fuzzy systems based on the interpretable knowledge can be initialized with some domain expert knowledge. There is removed the last layer performing the division thus the system has two outputs [1], [7] - [8]. The error on the first output will be computed taking into account desired output from learning data. Desired signal on the second output is constant and equals "one" (see Fig.1). After learning causes structures according to the initial idea, we can build the modular system. In such system the rules can be arranged in arbitrary order (see Fig.2). At the beginning of the numerical simulation, input fuzzy sets are determined by the modified fuzzy c-means clustering algorithm. Then, all parameters are tuned by the backpropagation algorithm. [1], [8] - [12].

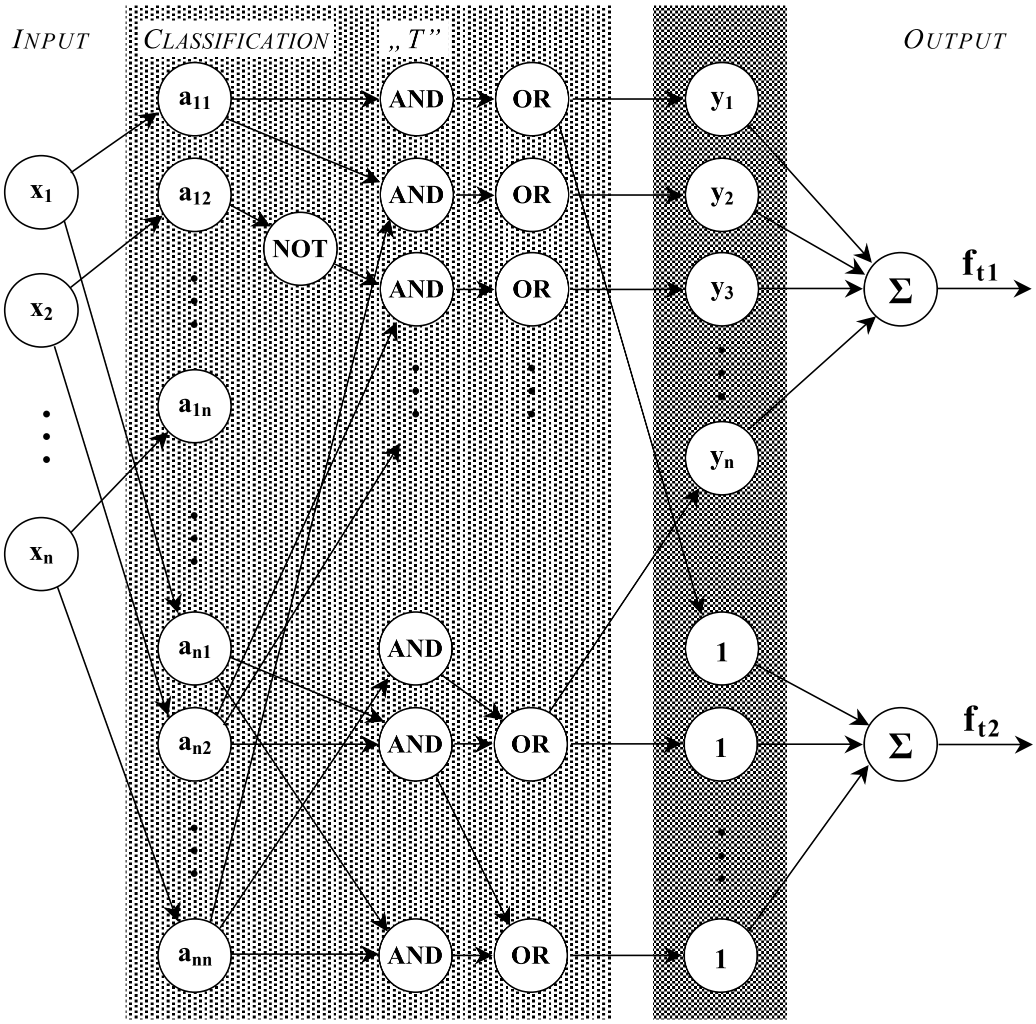

Fig. 1: A modified neuro-fuzzy system architecture

Solved network was trained and the related equations are presented below:

$$x(t) = [x_1(t), \dots, x_n(t)] \tag{1}$$

$$f(r_j(t)) = v_j(t) \tag{2}$$

$$f(e_j(t)) = y_j(t) \tag{3}$$

$$r_j(t) = \sum_{i=1}^{n+m} w_{ij}\, x_i(t) \tag{4}$$

$$e_j(t) = \sum_{i=1}^{k} w_{ij}\, v_i(t) \tag{5}$$

where x(t) is the input vector, $r_j(t)$ and $l_j(t)$ are input signals provided to the hidden and output layer neurons. Parameter k stands for the size of the hidden layers. Parameters $v_j(t)$ stands for the neurons activations in the hidden layer at time t, and $y_j(t)$ stands for the activations of the neurons in the output layer at time t, parameters w_{ij} are neurons weights values.

Mean squared error performance index is implemented during identification procedure:

ISBN: 978-988-17012-5-1

$$E_r = \frac{1}{t_{\max} - t_{\min}} \sum_{t=t_{\min}}^{t_{\max}} (y(t|t-1) - y(t))^2 \qquad (6)$$

where y(t|t-1) is the prediction for the sampling instant t calculated from the neural network model at the sampling instant t-1. The value y(t) is the real value of the process output variable. This value is collected during the identification procedure experiment. Variables t_{min} and t_{max} indicate range of the data used for identification procedure. If the performance index E_r is used for identification procedure, the obtained models are of better quality when one-step ahead prediction is necessary [1], [7], [12].

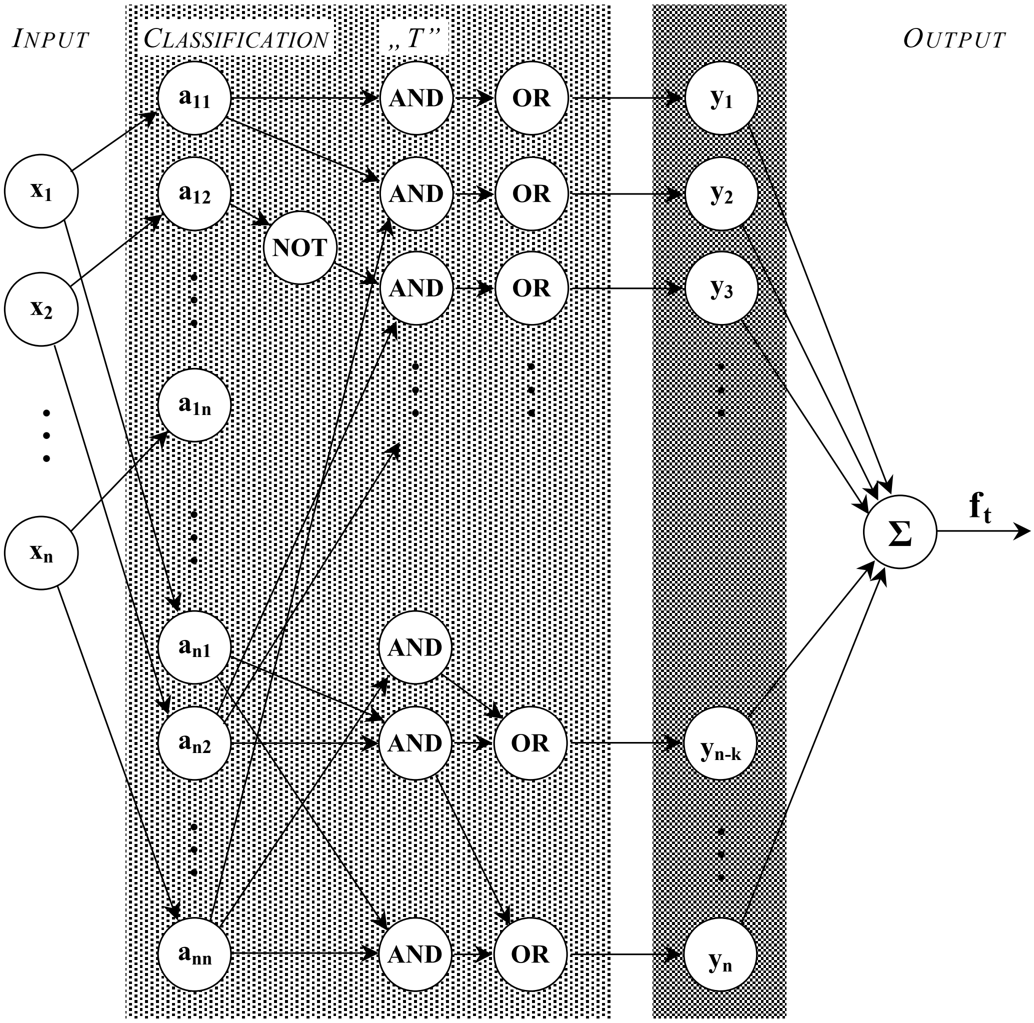

Fig. 2: A solved neuro-fuzzy system architecture after learning

The process under consideration within experimental implementation is diagnostic system for solved complex manufacturing processes control. For example, accuracy of the top composition model is studied. The fundamental model is used as the real process within identification experiment. We obtain two sets of data, namely training and test sets.

Neuro-fuzzy systems using can provide better results than the one based on traditional indices. Classifiers can be combined to improve network accuracy. Solved subsystems are trained by different datasets. There is also realized a detection in web implementation.

A. Experiments

The investigated problems were some well-known benchmark problems. The state classification problem of diagnostic parameters on base of measured forces and vibrations signals was solved. Within realized monitoring assignments, the tools states were to be classified into 4 classes, i.e. sharp tools, tools with an average wear of teeth of 0.2; 0.25 etc., and tools with broken (missing) insert. Seven features of the force components (F1 to F7) were selected using the solved feature selection methodology. Membership functions were respectively assigned to the input linguistic (imprecise, vague) variables. Corresponding to the described 4-class problem, the output variable (tool state) had 4 membership functions.

To solve these problems, the algorithm generally offers solutions with shorter and simpler rules sets and results in better recognition performance than the investigated previous neuro-fuzzy approaches. The experiments demonstrated the effective applicability of the proposed methodology.

All experiments were conducted using real and synthetic data collected from diagnostic problem solving statements. In experimental study, we evaluated the best parameters of neuro-fuzzy network and used learning algorithm. Results are comparable with the others best approaches. Experiments with web tools were focused first of all on structure effectiveness of used queries length. The achieved results confirm that the proposed methodology allows detecting the fault and identifying precisely the value of the faulty parameter in spite of disturbance incidence. The proposed method is modified to allow joining the rule bases into one fuzzy rule base.

III. An Internet Module

An approach for on-line parameters identification and measurements via internet technology uses the following designed structure of sentence transmitting, which represents some communication protocol between data server application and client application in Applet tool. We have:

//Received string structure
//[dt0.10][n0monitoringCourseName_vmin_vmax]0_51.33_4:5_22.0
//sample.timenum.graphNameofgraph_vmin_vmaxnum.graph_value.

An example of experiment approach within web tool is in this form [1] [12]:

1. We divide the domain queries set to 30 subsets with different length queries.
2. We train a network for each solved subset:

```
FOR i= 1 to 20
   FOR l= 1 to 19
      FOR m=1 to 18
               - - -
         choose randomly m/20 queries
         Network examination: against chosen queries
                              examine the used network
      END
   END
   EVALUATION of the rule: evaluate rule coefficients
END
```

3. We have a new domain queries set.

IV. Genetic Algorithm Methodology Approach

A measurement of the vibrations (temperatures and so on), analysis of input signals and their processing is an important part of this work. Sensors are connected to a processing unit. In the second phase we transform based signal features, which will be used as the inputs to domain knowledge-based system. There is the problem-dependent data structure representation, and cost function, i.e. fitness, evaluation, and the robust reproduction phase, which are functionally separated and may be common to each application. That's reason for creating a good and relevant genetic model for clustering and adopting efficient operators for the optimization [1].

A solved genetic algorithm starts to work with a set of domain knowledge structures that are coded into binary strings. The diagnostic rules, which are to be evaluated through genetic algorithm, should then be coded. Diagnostic rules are in the following form (production rules representation):

$$IF\ (S_1 \cap S_2 \cap ... \cap S_n)\ THEN\ (F_i) \tag{7}$$

This formula states that if symptoms S_1 to S_n are present then the i_{th} fault (F_i) occurs. The used symptoms S_1 to S_n correspond to n different on-line information sources, which could be on-line measurements and controller outputs. Each symptom is considered to take one of the following values: -*increase, steady, decrease, neutral.* Each symptom in the condition part of rule is coded by a *2-bit* binary, where „00“ stands for symptom *decrease*, „01“ stands for symptom *steady*, „10“ stands for symptom in*crease*, „11“ stands for symptom *neutral.* Symptom *neutral* means that the corresponding symptom is not important. For example, a *normal* rule applying can match with any values. By introducing this symptom, the condition parts of all the diagnostic rules will be of the same length and the corresponding parts of the rules will represent the same observations.

A technical computing environment is Matlab program tool. We define many parameters and operators within solved genetic algorithm approach. *Elitist component* decides if the best string generated up to time *t* should be included in the population of generation *t+1.* The distance between two strings is measured and their fitness is modified in the sense that strings in the same neighbourhood are forced to share their fitness among another, which effectively limits the uncontrolled growth of particular species within a population. We use three various crossover operators: *order crossover, cycle crossover and partially matched crossover*. We have also a *distance of crossover points* parameter, which determines the maximal distance between two crossover points in the used crossover operator for reordering problems. The *crossover unit* reflects the decision if the crossover operator (as usual) should consider genes as the smallest atomic entity. Another possibility considers if it should be applied such that logical subgroups of genes stay together as a structural unit [14], [16] - [20].

The crowding technology has been introduced to induce niche like behavior in genetic algorithm search in order to maintain diversity in the population. We realize the decision if the mutation amount of real-valued genes is determined according to a normal distribution or an exponential distribution. Special parameters determine the density function of the normal distribution of the mutation amount and the mean value of the exponential distribution of the mutation amount. *Mutation value replacement* reflects the decision of the mutation operator should overwrite the old gene.

V. Problem Solving By Means Of Chaos Theory Principles

Further, we analyze appropriate indicators of chaos approach, which are first of all fractal dimensions, metric entropies and Lyapunov exponents. The main property that makes them relevant in the description of attractors is their invariance under smooth changes of coordinates. Power spectra and auto-correlation functions are obsolete indicators and not allowing to distinguish between truly deterministic chaos and stochastic motion in this sense, that in both cases the spectrum is broad-band and the correlations decay exponentially. On the other hand, the asymptotic time decay of auto-correlation functions can be expressed, for piecervise linear maps, in terms of generalized Lyapunov exponent.

The concept of fractal dimension is related to information theory and refers to static properties of the invariant measure [13] - [15]. It provides a rough estimate of the number of degrees of freedom activity involved in the asymptotic motion. This fact indicates the possibility of constructing simple models, whenever a general theory is lacking.

A more appropriate indicator of chaos is given by the production of entropy associated with the motion. In strange attractors, at variance with ordered cases, the observation of increasingly long portion of trajectory allows to localize the initial condition with higher and higher accuracy. Such information is created at an exponential rate, named metric entropy.

We would like to find plausible uniformitarian mechanisms for evaluation of complex systems. The technology involves the appropriate combinatorial technique for improving the quality indices of the diagnostic signal parameters within genetic algorithm approach with respect to compression, performance speed, and the other significant characteristics of the model [17] - [19].

The modern combinatorial design techniques are well usable to coded design of signals and compression. However, the design based on the classical combinatorial theory approach is not always applicable because the problem is very often difficult to be solved by mathematical programming methods. So ingeniously ordered chain approach to the study of elements and events is known to be of widespread applicability, and has been extremely effective when applied to the problem of finding the optimum ordered arrangement of structural elements and bonds in a vector data coding system.

A general used formula for calculate of multidimensional vector data code is given by:

$$\frac{(k^2-k+1)}{R} = V_1 \times V_2 \times \cdots \times V_i \qquad with \quad (V_1, V_{2,} \quad \cdots \quad V_i) = 1 \tag{8}$$

We enumerate a set of nodes of $V_1 \times V_2 \times, \ldots, \times V_i$ (grid). Each node of the grid meet exactly R-times.

We consider fuzzy rule base of Takagi-Sugeno system form. The Takagi-Sugeno fuzzy model is described by a set of fuzzy implications, which characterize local relations of a system in the state space [1]. The main feature of this model is to express the local dynamics of each fuzzy rule, i.e. implication by a linear-state-space system model, and the overall fuzzy system is then modeled by fuzzy "blending" of these local linear systems models through some suitable membership functions. Approaches to chaotify discrete-time and continuous-time Takagi-Sugeno fuzzy systems are very different. For simplicity, we have a following illustrative example.

We consider a nonchaotic discrete-time Takagi-Sugeno fuzzy model, given by these rules:

$$IF\ x(t)\ is\ M_{f1} \quad THEN \quad x(t+1) = H_1\, x(t) + v(t) \tag{9a}$$

$$IF\ x(t)\ is\ M_{f2} \quad THEN \quad x(t+1) = H_2\, x(t) + v(t) \tag{9b}$$

with $x(t) \in [-b, +b], b > O$ with the membership functions M_1, M_2, where:

$$H_1 = \begin{bmatrix} b & 0{,}288 \\ 1 & 0 \end{bmatrix} \qquad H_2 = \begin{bmatrix} -b & 0{,}288 \\ 1 & 0 \end{bmatrix}$$
$$M_1 = \frac{1}{2}\left(1 - \frac{x(t)}{b}\right) \qquad M_2 = \frac{1}{2}\left(1 + \frac{x(t)}{b}\right) \tag{10}$$

The controlled Takagi-Sugeno fuzzy system is described as follows:

$$\begin{aligned} x(t+1) &= \sum_{i=1}^{n} \mu_i(t)\, H_i\, x(t)\, v(t) \\ &= \sum_{i=1}^{n} \mu_i(t)\, H_i\, x(t) + \delta \sin\left(\frac{\pi}{\delta} \beta\, x(t)\right) \end{aligned} \tag{11}$$

The controller is taken as a sinusoidal function. In the simulation, the magnitude of the control input is experimentally chosen to be $\delta = 0.09$. Thus, $\| v(t) \|_\infty < \delta$, and can also be regarded as a control parameter. Without control, the Takagi-Sugeno fuzzy model is stable.

Let the solved dynamical system is generally described by the differential equation [15] - [16], [19]:

$$x(t) \equiv \frac{dx}{dt} = F(x, \mu) \tag{12}$$

where x is a vector whose components are the dynamical variables of the system, $F(x,\mu)$ is a nonlinear function of x and μ stands for the control parameter. Let $x_1(\mu)$ and $x_2(\mu)$ be two different solutions of equation (12). A critical point is therefore defined by the equation :

$$x_1(\mu_c) = x_2(\mu_c) \tag{13}$$

which is an implicit equation for μ_c. The Jacobian matrix $G(x, \mu)$ associated with $F(x,\mu)$ is defined through:

$$F(x+\delta x, \mu) = F(x,\mu) + G(x, \mu)\, \delta x + 0(\delta x)^2 \tag{14}$$

The fundamental property that can be proved by using the implicit function theorem is:

$$det\, G(x, \mu_c) = O \tag{15}$$

This result implies critical slowing down. This is easily shown by a linear stability analysis of the solution $x_1(\mu)$.

We have the following assumption:

$$(x,\mu) = x_1(\mu) + \varepsilon x'(\mu)\, exp(\lambda t) + 0\, \varepsilon \tag{16}$$

The characteristic equation for λ is:

$$det\, \{\lambda I - G(x_1, \mu)\} = O \tag{17}$$

where I is the unit matrix, the ε-exponent is interpreted as a dimension. Hence at $\mu = \mu_c$ one root (at least) of (17) will vanish, implying an infinite relaxation time. This result remains true if $x_1(\mu)$ and $x_2(\mu)$ are time-periodic solutions of relation (12).

We realize the direct estimate of fractal dimension from experimental data (embedding theorem and related topics) with a particular attention to the effect of filtering on a chaotic signal.

All relevant quantities involved in the description of strange attractors require taking some limit (either in space or in time) [15] - [16], [19]. The fractal dimension, for example, is related to the scaling behavior of the (natural) invariant measure μ, when the observational resolution is increased. To be more specific, we first define a *partition* (covering) of the phase space, as a collection of disjoint open sets with variable size ε_i. In this way, we can associate a mass p_i to each element E_i of the partition as:

$$m_i = \int_{E_i} d\mu \qquad (18)$$

The mass m_i can be evaluated from the fraction of points belonging to E_i, when a sufficiently large number of the set points, is generated according to the measure μ. From the obvious consideration that $m \sim \varepsilon^1$ in the case of a plane, it follows that the ε-exponent is to be interpreted as a dimension. Accordingly, we can define a *local* dimension α_i in terms of size and mass of the *i-th* element of the covering:

$$m_i \sim \varepsilon_i^{\ \alpha i} \qquad (19)$$

where ε_i is assumed to be sufficiently small, and we let α_i explicitly depend on the index i. This is a crucial point that makes also a statistical approach most appropriate. Metric entropy using improves this approach. It is also worth noting that such a definition of dimension is not affected by the existence of multiplicative factors, since they only yield corrections to the leading scaling behavior. Hence, they can, in principle, be neglected, although their contribution is often relevant in numerical simulations. The distinctive features of a numerical algorithm for the estimation of fractal dimensions are essentially related to the partition used to cover the set. Therefore, the task starts from the rules to generate a suitable covering (according to the probability of each element). This approach can be extended to metric entropies and Lyapunov exponents for problem solving improving.

A more appropriate indicator of chaos is given by the production of entropy associated with a motion. In strange attractors, at variance with ordered cases, the observation of increasingly long portions of trajectory (suitable for solved diagnostic signal analysis) allows to localize the initial condition with higher and higher accuracy. Such information is created at an exponential rate, i.e. metric entropy.

We start with metric entropies which are defined in terms of a sequential measurement, i.e. a series of observations of a trajectory at equally spaced times ($t_n = n\Delta t$, in the case of continuous time, and $t_n = n$ for maps). To be more specific, we consider a discrete-time dynamical system. We have the mapping that is strictly deterministic. We have the initial condition x_o with infinite precision, the trajectory $x_n = F^n(x_o)$ is uniquely determined. Let us also assume that an observation of the system be done with limited resolution. Therefore, we introduce a partition of the phase-space in L regions E_j. When the representative point x_n is in the element E_j, the *reading instrument* displays the value j. In this way, a symbolic sequence $S_N = \{s_n\ ,\ n=1,\dots,\ N\}$ can be associated to each trajectory x_n, $n=1,\dots, N$ (for finite N, S_N is also named word). The symbol $s_n \in [1,L]$ represents the index of the partition element visited by the trajectory at time n.

The order of symbols in the word is crucial in determining the metric entropy. The conditional probability of being in subset E_{ij} at time n, given an initial condition in E_i, is:

$$p(j,n|i) = \frac{\mu(F^n\, E_i \cap E_j)}{\mu(E_i)} \qquad (20)$$

where μ is the invariant measure. The mass $\mu(E_i)$ is the same as that contained in F^nE_i , and the conditional probability is normalized in such a way that

$$P\ (j,0\ |\ I)\ =\ \delta_{i,j} \qquad (21)$$

We have for a mixing system:

$$\lim_{n\to\infty} p(j,n|i) = \mu(E_j) \qquad (22)$$

That is, the image of the initial set covers the whole attractor and no correlation survives between arrival and starting element. The action of the map F in phase-space is translated into a shift of symbols in the associated space:

$$x_n \longrightarrow x_{n+1} \Rightarrow \{\dots,\ s_{n-1}, s_n,\ s_{n+1},\dots\} \longrightarrow \{\dots,\ s_n, s_{n+1}, s_{n+2},\ \dots\} \qquad (23)$$

The time origin in the (doubly-infinite) symbol sequence is moved one place to the right. When a generating partition is not known, as in the case of experimental data or in most of computer simulations, it is possible to divide the searching space of size $\varepsilon_i < \varepsilon$ (they are usually taken all equal for simplicity) and evaluate the *metric entropy* as:

$$K(q) = \lim_{\varepsilon\to 0}\ \lim_{N\to\infty} \frac{1}{N}\frac{1}{(1-q)} \ln \sum_{S_N} p^q\ (S_N) \qquad (24)$$

where the limit $\varepsilon \to 0$ guarantees that a generating partition is finally obtained. The performed analysis assumes implicitly the knowledge of all coordinates of each attractor's point in phase space. This is certainly the case of all numerical simulations, but it is not always possible in experiments. Sometimes, just one variable can be measured at different times.

In this way, we construct the values and parameters of genes within solved domain chromosome. We solve metric entropy for each element or for solved chromosome within genetic algorithm running. This chromosome implementation by genetic algorithm approach is near to real diagnostic situation representation. Nowadays, some experiments were realized. Achieved results are very promising.

VI. Fitness Function

We realize a fitness function evaluation. The fitness *FIT* of the rule is calculated by following way [1], [12], [21], [22]:

$$FIT = \frac{k_1\, O}{k_2\, P + 1} + k_3\, Q \tag{25}$$

where O ....is a number of successful applications of a rule when tested by the training data corresponding to this rule,
P.............is a number of incorrect applications of the rule when tested by the rest of the training data,
Q ....is a number of *neutral's in* the condition part of the rule.
k_1, k_2, k_3...are positive weighting coefficients,

The requirement that *FIT* should not be negative is determined by the genetic algorithm, which is used here in solved problem. In the case of the reproduction phase, an individual rule is selected with a probability, which is equal to the ratio of the fitness of the rule and the sum of the fitnesses of all the rules in the generation. This is also the reason for conclusion, that the fitness of a rule should not be less than zero.

This fitness function (25) is not a linear function. By inspecting the first order partial derivatives, if we suppose that *O, P, Q* could change continuously, it is found that the fitness function (25) is preferable to the first one. Differentiating fitness function (25) provides following equations:

$$\frac{\partial\, FIT}{\partial\, O} = \frac{k_1}{\partial\, P + 1} \tag{26a}$$

$$\frac{\partial\, FIT}{\partial\, P} = \frac{-k_1 k_2 O}{(k_2 P + 1)^2} = \frac{-k_2 O}{k_2 P + 1} \cdot \frac{\partial\, FIT}{\partial\, O} \tag{26b}$$

$$\frac{\partial\, FIT}{\partial\, Q} = k_3 \tag{26c}$$

A non-linear fitness function is more sensitive to changes in *O, P* when a rule is close to the desired rule, with large *O* and small *P*. The values of the parameters k_1, k_2, k_3 in relation (25) can affect the result of learning. The choices of k_1, k_2, k_3 are determined by the quantity of training data and the user's objectives.

Numerical results verify the theoretical analysis and the design of the proposed chaos generator with diagnosis data applying. This approach was focused on exploring the relation between fuzzy theory and chaos theory, and combines fuzzy and chaos control technologies with metric entropy for practical applications of solved diagnostic predictive system

A cardinal rule of this approach is illustrated in Fig. 3.

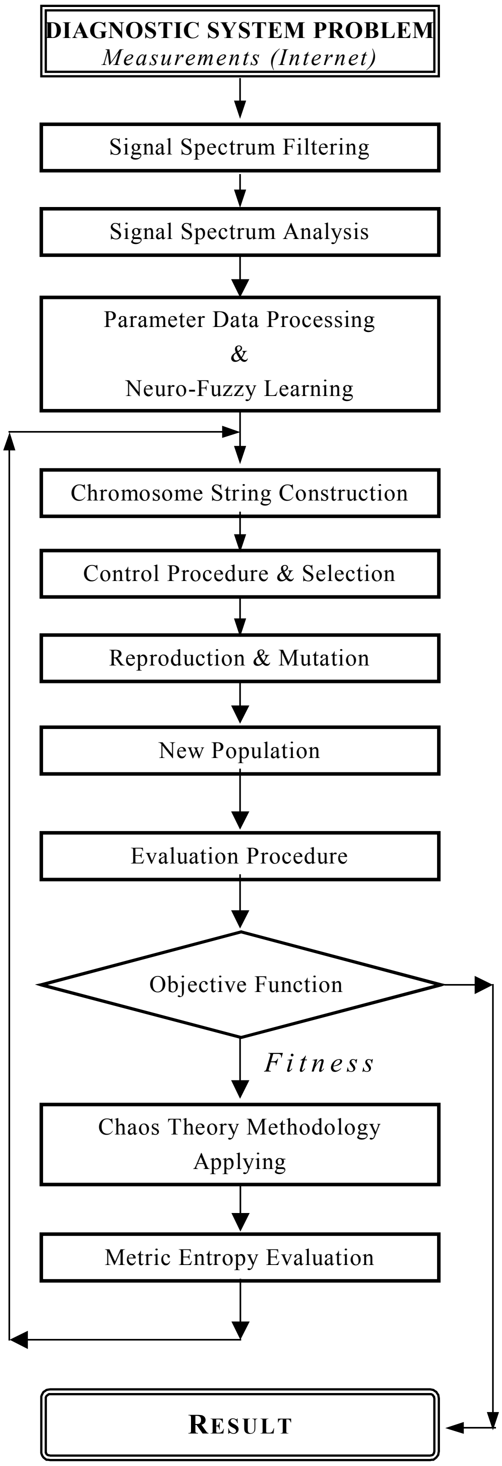

Fig. 3: Optimization problem architecture.

VII. Obtained Results

The Most Significant Decisions Indicators are illustrated in Fig.4.

A remarkable property of the model is an ability to reproduce the maximum number of combinatorial varieties in the systems with a limited number of elements and bonds. Among the experimental variants provided for the diagnostic system problems solving methodologies, the proposed approach clearly dominates the other variants. It provides qualitatively higher performance level of implementation. Practically, there are no instable parameters courses. This is graphically illustrated also in Fig.5.

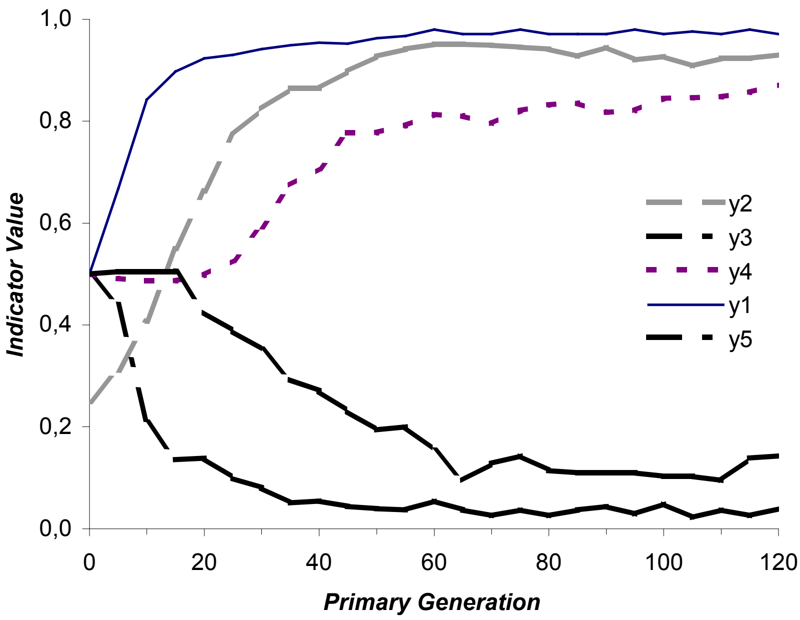

Fig. 4: The most Significant Decisions Indicators:
y1 = crossover units course, **y2** = selection method course, **y3** = elitist model course,
y4 = mutation function course, **y5** = value replacement course

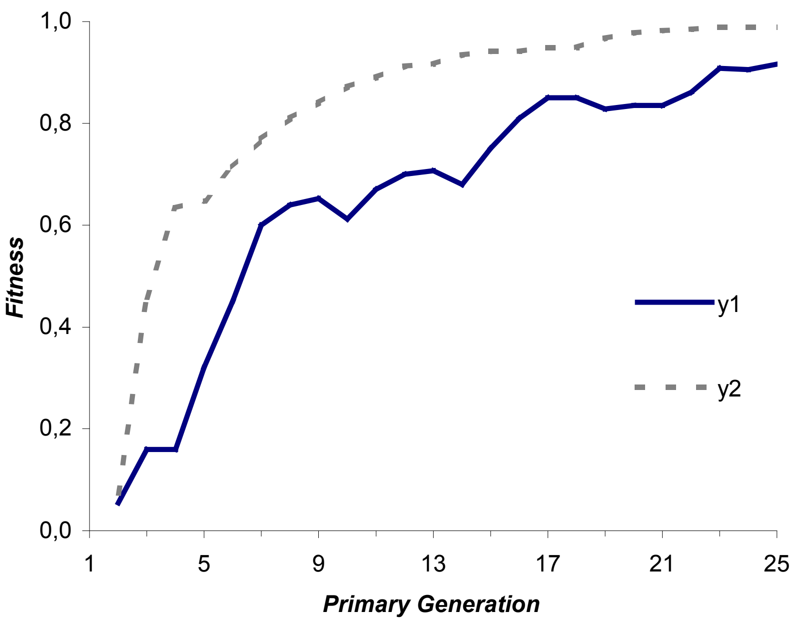

Fig. 5: Fitness Evaluation Course:
y1 = Fitness curve for the classical genetic algorithm approach
y3 = Fitness curve for the proposed methodology with chaos theory principles applying

Fitness function is near to the best solution (as expected) within solved phase of run. The solved approach requires the more complex real-valued gene coding.

This mechanism with the strength and specificity rules management can be effectively assimilated to a genetic operator. So it may be interesting to compare this solution with above mentioned genetic algorithm approach.

The learnt rules have been tested on the real process. All simulated faults were successfully diagnosed by the corresponding rules and no incorrect diagnosis occurred.

A genetic algorithm has to maintain a balance between the preservation of good combinations of genes, and the exploration of new combinations. We adopt a successful strategy for achieving this balance which has been to combine a highly explorative, or disruptive crossover with elitism, in which a fraction of the best individuals found so far survive into the next generation. Elitism gives better individuals more chances of mating to produce fit offspring, an advantage when their offspring will frequently be poor. We compare *Fitness evaluation* for classical (without chaos theory principles applying) approach and solved methodology approach. Judged input parameters for experimental simulations have been the same for both methodological cases [1], [17]. The result is illustrated in Fig. 5.

VIII. CONCLUSION

Solved neuro-fuzzy system seems to perform both the fundamental requirements of intelligent manufacturing, i.e. real-time nature, uncertainties handling, learning ability and managing both symbolic and numerical information, and the expectation to generate sufficient rule set also for larger problems, which would be handled by usual neuro-fuzzy models only with severe difficulties.[1], [12], [19]. The solved system is a good illustration of how a genetic algorithm approach to a complex practical combinatorial problem can provide an extremely robust solution with several practical advantages. The main difficulty in the problem is that there typically is a multitude of local extreme, which happen to be located close to a bounding constraints, conventionally imposed at the given threshold, and that anyway has to be imposed out of safety considerations. From this point of view, the proposed methodology achieves the best results.

The aim of the research described here was to investigate the factors involved in designing a genetic algorithm with respect to the overall objective of robustness and utility as a practical tool. By robustness, we mean the ability of the program to produce good solution in reasonable time, independently of any user interaction in the form of careful set-up or run-time intervention, possibly requiring considerable expertise. The rules could be improved by including additional features, such as the magnitudes of deviations in measurements, in their condition parts to increase their resolution.

There is some possible future extending. The fitted structures are selected and combined in a structured yet randomised way to produce more fitted structures, whereas, in a combinatorial search, all the possible structures will be evaluated with the same possibility. From the achieved experimental results, a promising performance of solved genetic learning approach can be expected when it is applied to more complicated tasks.

We solve also genetic algorithm methodology applications to the self-learning of diagnostic rules for industrial processes. Self-learning of diagnostic rules can facilitate knowledge acquisition effort and is more desirable in these cases where certain knowledge is unavailable.

The solved genetic algorithm approach is feasible by implementing it in a multi-transputer environment. Performance results for finding the best genetic algorithm for the problem of optimal operation parameters in solved diagnostic system have demonstrated the quality of our implementation.

Future research will be focused first of all on improving the runtime performance of solved implementation, including other genetic operators in the architecture and investigating the results of further test problems in more detail. There are will be investigated self-learning approaches of diagnostic rules through more advanced genetic and evolutionary algorithms and modified chaos theory principles. Further research will be also focused on exploring the relation between fuzzy logic approach and chaos theory, and combining fuzzy and chaos control technologies for real applications. Nowadays, some achieved results seem to be very interesting.

We realize the direct estimate of fractal dimension from experimental data (embedding theorem and related topics) with a particular attention to the effect of filtering on a chaotic signal. Genetic algorithm will have been the main rule discovery algorithm. It will concern to obtain self-organisation of a kind of communication protocol among a solved population.

The research reported on this paper is supported by following scientific project of Slovak Grant Agency VEGA (Ministry of Education): *Intelligent Approach to Automated Diagnostic Problem Solving of Machinery Equipment* (Number: VEGA 1/4133/07).

REFERENCES

[1] Gallová, Š.: Some effective techniques applying for complex diagnostic problem solving via genetic algorithm approach. In: Computational Intelligence: Methods and Applications. [Warsaw] : EXIT, 2008. p. 183-207. ISBN 978-83-60434-50-5.
[2] Ballé, P.: Fuzzy Model-based Parity Equations for Fault Isolation. Control Engineering Practice, No7, pp.261-270, 1999
[3] Zhang, J, Roberts, P. D.: On-line Process Fault Diagnosis Using Neural Network Techniques. Institute of Measurement and Control, 1992
[4] Montana, D., Davis, L.: Training feedforward Neural Networks using Genetic Algorihms. Proceedings: 11th Int.Joint Coference on Artificial Intelligence, pp.762-767, 1989
[5] Goldberg D. E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, reading, Massachusets, 1989
[6] Abraham, N. B.:, Lugiato, L. A; Narducci, L. M.: Instabilities in Active Media. J.O.Society Am.B.;1999
[7] Korytkowski M., Rutkowski L., Scherer R.: On Combining Backpropagation with Boosting. In: International Joint Conference on Neural Networks, IEEE World Congress on Computational Intelligence, Vancouver, Canada, 2006
[8] Wang L.: Adaptive Fuzzy Systems and Control. PTR Prentice Hall, Englewood Cliffs, 1994
[9] Mrugalski M.: Neural Network based modelling of non-linear systems. University of Zielona Gora, Poland, 2004
[10] Patton R.J., Chen J.: On eigenstructure assignment for robust fault diagnosis. Journal of Robust and Nonlinear Control. 10/11.,pp.1193-1208, 2000
[11] Scola H.R., Nikoukah R., Delebecque F.: A linear programming approach. Journal of Applied Mathematics and Computer Science.No13, pp.515-526, 2003
[12] Gallova S.: An Effective Approach to Mamdani neuro-fuzzy System Problem Solving. Transactions of the Universities of Kosice. Research Reports of the Universities of Kosice, 4/2008, pp.51-55.
[13] Mandelbrot, B. B.: The Fractal Geometry of Nature. Freeman, San Francisco, 1982
[14] Li, Z; Park, J. B.; Joo, I. H.; Chen, G.; Choi, I. H.: Anticontrol of Chaos for Discrete TS Fuzzy Systems. IEEE Trans., Circ.-Syst.1, Vol.49, 2/2002
[15] Mandel, P.; Erneux, T.: Dynamic versus Static stability. Frontiers in Quantum Optics. A.Hilger, Bristol, Boston, 1986
[16] Badii, R.; Politi, A.: Strange Attractors. Phys.Letters, 104A, 303, 1984
[17] Gallová, Š.: Fault Diagnosis of Manufacturing Processes via Genetic Algorithm Approach. In: IAENG Engineering Letters, Volume 15 Issue 2, 2007. Pages 349-355. ISSN: 1816-093X.
[18] Gallová, Š.: Fuzzy Ontology and Information Access on the Web. In: IAENG International Journal of Computer Science, Volume 34 Issue 2, 2007. Pages 234-238. ISSN: 1819-656X.
[19] Gallová, Š.: A maximum Entropy Inference within Uncertain Information Reasoning. In: Information Processing and Management of Uncertainty in Knowledge-based Systems: Proceedings: Paris, Les Cordeliers, July 2-7, 2006. Paris: E.D.K., 2006. p. 1803-1810. ISBN 2-84254-112-X.
[20] Zadeh L.: The Concept of Linguistic Variable and It´s Application to Approximate Decision Making. Moscow. Mir, 1976.
[21] S. A. Goldman, R. L. Rivest: A Non-iterative Maximum Entropy Algorithm. In: Koval, L. N. & Lemmu, F. J. (eds.) Ucertainty in Artficial Intelligence, Vol.2, pages 133-148, North-Holland, 1988.
[22] R. W. Hamming, Coding and Information Theory. Prentice Hall Inc., Englewood Cliffs, N.J., 1980