Genetic Algorithm as a Tool of Fuzzy Parameters and Cutting Forces Optimization

Stefania GALLOVA, *Member, IAENG*

***Abstract*—The classification of solved signal features for manufacturing process condition monitoring has been carried out using fuzzy parameters optimization processing. In cases where assumptions in respect of nonlinear behavior cannot be made, the need to describe mathematically, ever increasing complexity become difficult and perhaps infeasible. The optimization possibilities of the fuzzy system parameters using genetic algorithms are studied. An analytical function determines the positions of the output fuzzy sets in each mapping process, that substitute the fuzzy rule base used in conventional approach. We realize case adaptation by adjusting the fuzzy sets parameters. Fuzzy parameters within optimization procedure could be multiobjective. We solve also the system for cutting process simulation, which contains the experimental model and the simulation model based on genetic algorithms. There is developed a genetic algorithm based simulation procedure for the prediction of the cutting forces. These genetic algorithms methodologies are suitable for fuzzy implementation control and for solution of large-scale problems.**

***Index Terms*— cutting forces optimization, fuzzy parameter optimization, fitness, genetic algorithm.**

I. Introduction

We evaluate cutting conditions subjected to various machinery and manufacturing constraints to ensure the process quality and process efficiency. The operation of the cutting forces simulation model can be confirmed by experimental results. The solved problem applies also fuzzy set theory mathematization approach. Stable cutting experiments are defined by straight cutting using parameters of technology. Unstable cutting experiments consist of different causes of instabilities. The realized simulation process is based on genetic algorithm and on the analytical formulation of the cutting process components. Genetic algorithm may be applied generally without recourse to domain-specific heuristics. Compared to traditional simulation process and optimization methodologies, a genetic algorithm is global and robust. We use some artificial intelligence tools, such as expert system, genetic algorithm methodology and fuzzy set theory principles. These approaches cooperate each other. We use also progressive internet technology tools. The optimization and the simulation of machining parameters and cutting forces become easier. The configuration of combination of genetic algorithm methodology with expert system approach and web technology is illustrated in Fig.1.

The classification of solved signal features for manufacturing process condition monitoring has been carried out using fuzzy parameters optimization processing. We realize the transition in computer applications from *data processing* to *information processing* and then to *knowledge processing.* The genetic algorithm based simulation procedure is proposed to predict cutting forces. The procedure evaluates the cutting conditions subjected to constraints such as cutting speed, cutting width, feeding and cutting depth.

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 in the condition part of rule is coded by a *2-bit* binary. The use of strong and weak functional dependencies between sets of domain fields to factor the product into a family of projections in *„normal form“* allows entry and update requests to be treated as single changes in projections that together generate a multiple change in the database state that is legal and preserves integrity.

If we consider the used database as an system whose inputs consists themselves of positive and negative subsets of *„I“* (i.e. items to be added to, and taken from, the database), three types of illegal transitions or subtractions may be requested:

1. Additions or subtractions to the state subset that lead to an impossible state subset such that no further additions or subtractions can lead to a legal state. The change must be rejected for integrity.
2. Additions such that there exists a unique minimal subset of further additions or subtractions leading to a legal state. This case allows a unique completion of the changes that preserves integrity.
3. Additions or subtractions such that there exist several incomparable minimal subsets of additions or subtractions leading to a legal state. This case may be regarded either as offering a choice of completion or necessitating rejection because of its indeterminacy.

The normal forms are important because the database projections involved give a simple model of the database structure – as the sum of its projections – has pointed out how these projections serve as a *„model of the organization“* [1], [2].

We determine the degree of possibility that a data structure (or a family of data structures) is contained in (implies) the data structure (or the family of data structures). Fuzzy relation

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).

theory is the proper mathematical implement to be used for the structural analysis of the real-world data. The investigation of various aspects of this theory has some conclusions. The choice of the semantic properties of a particular operator of implication procedure depends on some pragmatic consideration determined by the methodological questions of a particular application. We are forced to determine the value of the membership function of the possibility distribution in the real-world applications. This necessitates introducing an observer as estimator of the membership function. There exist many distinct meaningful ways of defining the containment of one fuzzy structure in another. These depend on the choice of a particular implication operator for the fuzzy power-set theory in hand.

Expert system environment is used as an intelligent adviser for users. It has a large domain knowledge base that is usable for genetic algorithm procedures.

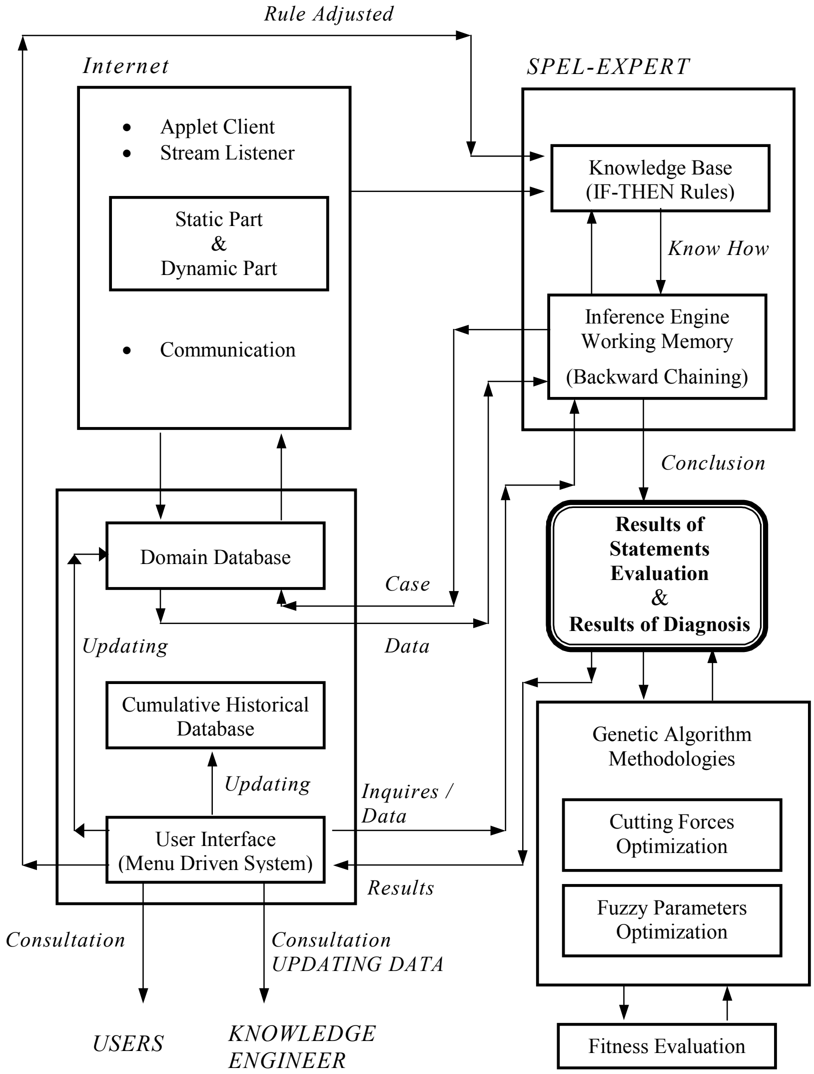


Fig. 1. The architecture of problem solving.

II. Genetic Algorithm Optimization Methodology Approach

The genetic algorithm used here is the three operator approach with some modifications. We explicitly distinguish between decisions and parameters values. In the absence of an a priori known best or worst fitness of a string, the solved methodology of fitness calculation is based on the idea of measuring the increase of fitness created by the secondary genetic algorithm approach. We have a parameter $F^{m}{}_{n}$, which is the best fitness of a string in the secondary population for test problem example *m* after generation *n*. A parameter p_t is the number of test problems. The *fitness* of the secondary genetic algorithm GA_{Fs} after generation *n* is given by [3], [4]:

$$Fitness\left[GA_{FS}\right] = \frac{1}{p_t}\sum_{m=1}^{p_t}\frac{\max(0, F_n^m - F_0^m)}{1 - F_0^m} \qquad (1)$$

By this way, we will avoid distorted fitness values arising from possibly different degrees of complexity of test problem examples. The fitness of an individual structure is a measure indicating how fitted the structure is [5]. We realize many experiments with various changes of genetic algorithms operators. For example, we implemented mutation as follows. A single mutate step with two new (created) children consists of randomly choosing two bits in the string of length 1,71 and interchanging their values. The probability that a single mutate will actually modify an individual is 0.29. The performance profile with indicators of the most significant decisions become important after the 6Oth primary generation, when other decisions and parameters have already been appropriately determined. The optimal crossover probability evolved as 0.52. This value remained nearly constant for all strings lengths investigated. The optimal mutation probability value increases with increasing numbers of genes in a string.

There is realized genetic *fuzzy rule learning*. The goal of this algorithm approach is also to find an effective fuzzy *If-then-else* rules to predict the class of input patterns correctly. The proposed learning approach is performed on intrusion detection that is a complex classification problem [1], [2], [7], [8].

We produce an initial population. Each individual in the population represents of fuzzy *IF-THEN-ELSE* rules in the population. The genetic fuzzy system is considered for each of the classes of the classification problem separately. Benefits of this separation are also that the learning system can focus on each of the classes of the classification problem. The mentioned random is extracted according to the patterns of the training dataset, which their consequent class is the same as the class that the algorithm works on. There is determined the most compatible combination of antecedent fuzzy sets using the six linguistic values (see Fig.2).

There is calculated the compatibility of antecedent fuzzy rules with the random pattern. The final classification system is capable of detecting known intrusive behavior in a computer network. Our experiments show that it has an acceptable performance.

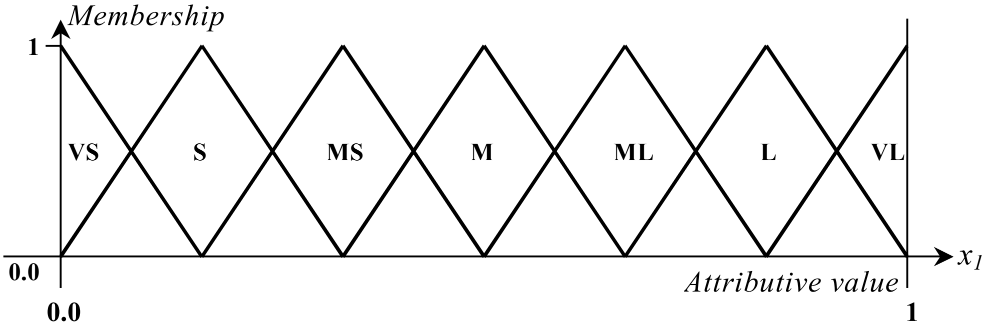


Fig. 2. Membership Function of Seven Linguistic Variables
VS=**V**ery **S**mall; **S**=**S**mall; **MS**=**M**edium **S**mall;
M=**M**edium; **ML**=**M**edium **L**arge; **L**=Large

To assign fitness to the evaluated fuzzy rule, the rule is first combined with the best rule in the other population using the following template:

if Rule1 $\geq$ 0.6
then high-permeability
else if Rule2 $\geq$ 0.6
then low-permeability
else medium-permeability,

where *Rule1* is a rule from the first population and *Rule2* is a rule from second population. If the evaluated rule is from the first population, the best rule from the second population is used to complete the template. If the evaluated rule is from the second population, the best rule from the first population is used to complete the template. This combined *if-then-else* rule is then tested on the training data and the interpretation results are compared with the transformed permeability. If the *if-then-else* rule gives the correct interpretation, it is a hit. The percentage of the hit among the training data is the fitness of the evaluated rule. To provide shorter and more readable rules to be evolved, rules with length more than 90 nodes are penalized. Also the *best* rule in end population is updated at the beginning of every generation, so that a good rule can be immediately used to combine with rules in the other population and input the evolution.

To understand why rules that classify high-permeability segments first have produced better results, we calculated population fitness and the fitness of the best solution for all runs.

The crowding technology has been introduced to induce niche like behavior in genetic algorithm search in order to maintain diversity in the population (diversity necessary for continuous evolution). Experimental results from runs where population 1 evolves rules to identify high-permeability data.

When the first population is used to evolve rules that classify low-permeability segments and the second population is used to evolve rules that classify high-permeability data segments, Co-evolution pressure biased toward the second. Average fitness of the first population is consistently lower than that of the second population. Using the worse of the two rules (the one from the first population) as *Rule1* to interpret permeability has impaired the overall interpretation accuracy. The bias, however, does not appear in the other experiment where the rules that classify high-permeability were used as a *Rule1* to interpret permeability.

Both populations co-evolve together with comparable average fitness. This is a healthy co-evolutionary dynamics that has produced combined if-then-else rules that give more accurate permeability interpretations than that by other experiment. In both sets of experimental runs, the two populations improved very quickly at the first 190 generations. After that, the improvement is not very visible. This pattern also appears in the fitness improvement of the best combined overall permeability interpreter although to a lesser extent. One possible mean is that the best solution used to combine with individuals in the other population. 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 [6] - [8].

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.

III. Synthesizing Process Of Analytic Fuzzy Logic Controller

The process of *synthesizing analytic fuzzy logic* controller can be described as follows:

1. We define a new adaptive shapes and distribution of the input fuzzy sets by fuzzyfication interface [14], relations (2a), (2b).
2. We calculate the analytic calculation function f_1, for the activation of corresponding j-th output fuzzy set with degree $f_{j.}$ (3).
3. We initiate an analytic function that maps the values of input variables into positions of the centers y_{cj} of the corresponding output fuzzy sets (inference engine) (4).
4. We use analytic expression for the determination of the output value from the analytic fuzzy logic controller, as function of the shapes of the output fuzzy sets and their corresponding positions of the centers and activation functions (deffuzyfication interface) (5).

$$M_i^j(x_j) = \frac{1}{2^{\left(B_j \cdot e_i^j |x_j|\right)}} \left\{ \frac{1 - \cos\left[2\pi\, e_i^j (x_j - x_{ci}^j + \frac{T_i^j}{2}\right]}{(e_i^j - 1) T_i^j} \right\} \tag{2a}$$

with conditions:

$$-1 \le x_j \le x_{ai}^j \qquad x_{ai}^j \le x_j \le x_{bi}^j \qquad M_i^j(x_j) = 1 \qquad x_{bi}^j \le x_j \le 1$$

$$M_i^j(x_j) = \frac{1}{2^{\left(B_j \cdot e_i^j |x_j|\right)}} \left\{ \frac{1 - \cos\left[2\pi\, e_i^j (x_{ci}^j + \frac{T_i^j}{2} - x_j\right]}{(e_i^j - 1) T_i^j} \right\} \tag{2b}$$

$$f_j = \sum_{i=1}^{n_j} M_i^j(x_j) \qquad j = 1, 2, \dots, m \tag{3}$$

$$y_{cj}(t) = V_m F_j \left(1 + |x_j(t)|\right)^{A_j} \left[1 - \frac{f_1(t)}{n_j(t)} \operatorname{sgn}(x_j(t))\right] \tag{4}$$

$$v(t+1) = \frac{\sum_{j=1}^{m} \left(f_1(t)\, y_{cj}(t)\, T_j \frac{e_{oj} + 1}{2 e_{0j}} \right)}{\sum_{j=1}^{m} \left(f_j(t)\, T_j \frac{e_{0j} + 1}{2 e_{0j}} \right)} \tag{5}$$

where $M_i^{j\,(x_j)}$ is the i-th membership function for the j-th input variable, x_j *is normalized j-th input variable,* e_i^j (i=1,n) and B_j (j=1,m) adaptation and distribution parameters, respectively for the j-th input fuzzy set, V_m is the maximal value of control variable v, both F_j and a_j are free parameters, y_{cj} (t), T_j and e_{oj} is the centre, base and adaptation parameter of the j-th output fuzzy set, respectively.

The aim is to detect changes of the current process behavior and to generate analytical symptoms. The diagnosis task is accomplished by fuzzy evidential approximate reasoning scheme to handle different kinds of uncertainty that are inherently present in many real world processes, and to make decision under conflicting data or knowledge. The solved diagnostic system serves several purposes. It identifies the optimal decision boundaries between the different faulty states with as many details as possible or needed even in the presence of noise and uncertainties.

If an appropriate structure is identified, the learning task can be accomplished by any suitable training algorithm such as the classical backpropagation algorithm [2], [4], [9].

IV. Parameter Optimization Procedure

Procedure of parameter optimization is realized by following way. The identification procedure of obtained parameters leads to optimization and tuning procedures. We solve the forward procedure. The functional models FM_{ij}, $i=1,\dots,m;\ j=1,\dots,l$ are identified by solving a least square problem. The backward procedures fix the functional models. The parameters of the membership functions p_{ik}, q_{ik}; $i=1,\dots,m;\ k=1,\dots,n$ are updated by an effective non-linear gradient descent optimization technique. It requires the computation of derivates of the objective function to be minimized with respect to the parameters p_{ik}, q_{ik}. We apply the optimization algorithm with variable learning rates process using.

We have a set $S = (x^p, s^p)^N_{p=1}$, such that $x^p \in X \subset R^r$; $s^p \in Y \subset R^l$, the objective is to find subsystem $y_j(x^p)$ in the form:

$$y_j(x) = \frac{\sum_{i=1}^{m} FM_{ij} \prod_{k=1}^{n} e^{-\frac{(x_k - p_{ik})^2}{q_{ik}^2}}}{\sum_{i=1}^{m} \prod_{k=1}^{n} e^{-\frac{(x_k - p_{ik})^2}{q_{ik}^2}}} \tag{6}$$

We minimize the function of the mean squared error:

$$E_r = \frac{1}{2} \sum_{j=1}^{l} (y_j - s_j^p)^2 \tag{7}$$

where $x^p \in S$.

We solve the mean p_{ik} , variance q_{ik} (i.e. ellipsoidal functions) and the adjustment of the FM_{ij}. We also assume that $p_{ik} \in X_i;\ q_{ik} > 0;\ FM_{ij} \in Y_j$. We solve a complex non-linear multi-input and multi-output relationship with $x=(x_1, x_2,\dots,x_n)^T \in X \subset R^n$. Parameter x is the vector of input variables. We have also $y \in Y \subset R^l$. Parameter y is the vector of output variables. Output y and E_r depend on p_{ik}, q_{ik} only through equation (6). We have the following equations:

$$y_j(x) = \sum_{i=1}^{m} FM_{ij} \frac{\prod_{k=1}^{n} e^{-\frac{(x_k - p_{ik})^2}{q_{ik}^2}}}{\sum_{i=1}^{m} \prod_{k=1}^{n} e^{-\frac{(x_k - p_{ik})^2}{q_{ik}^2}}} \tag{8}$$

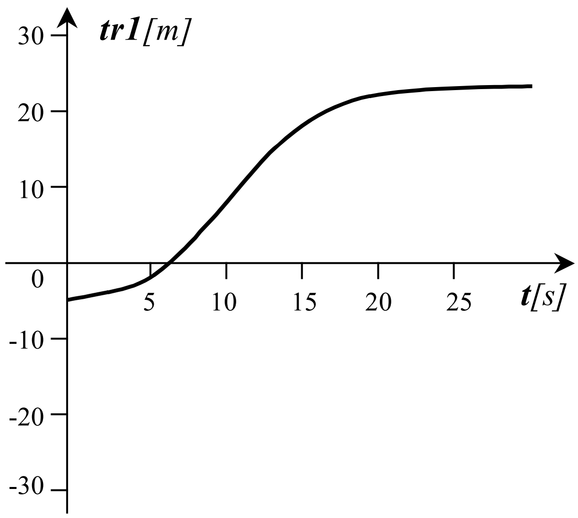
We realize derivatives of E_r with substitution $U = \prod_{k=1}^{n} e^{-\frac{(x_k - p_{ik})^2}{q_{ik}^2}}$:

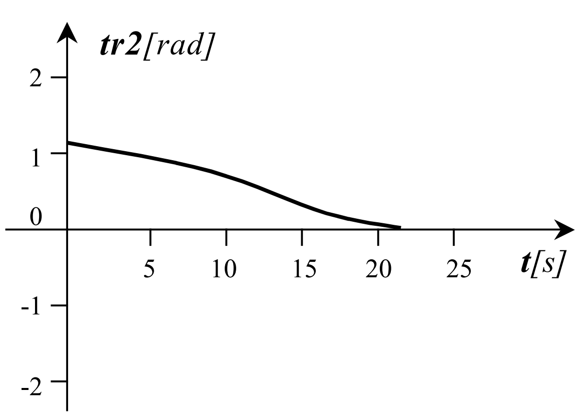
$$\frac{\partial E_r}{\partial p_{ik}} = \frac{\partial E_r}{\partial U} \cdot \frac{\partial U}{\partial p_{ik}} = \sum_{j=1}^{l} \left(\frac{\partial E_r}{\partial y_j} \cdot \frac{\partial y_j}{\partial U} \right) \cdot \frac{\partial U}{\partial p_{ik}} = \left[\sum_{j=1}^{l} \frac{(y_j - s_j) \cdot (FM_{ij} - y_j)}{\sum_{i=1}^{m} U} \right] \cdot \left[2 \cdot U \cdot \frac{(x_k - p_{ik})}{q_{ik}^2} \right] \tag{9}$$

$$\frac{\partial E_r}{\partial q_{ik}} = \frac{\partial E_r}{\partial U} \cdot \frac{\partial U}{\partial q_{ik}} = \sum_{j=1}^{l} \left(\frac{\partial E_r}{\partial y_j} \cdot \frac{\partial y_j}{\partial U} \right) \cdot \frac{\partial U}{\partial q_{ik}} = \left[\sum_{j=1}^{l} \frac{(y_j - s_j) \cdot (FM_{ij} - y_j)}{\sum_{i=1}^{m} U} \right] \cdot \left[2 \cdot U \cdot \frac{(x_k - p_{ik})^2}{q_{ik}^3} \right] \tag{10}$$

Above-mentioned optimization procedure of parameters that is obtained by the identification procedure uses an effective training methodology. The used learning process is performed in two stages. A clustering algorithm, first of all, finds a course model that roughly approximates the underlying input-output relationship. Then the procedure of parameter optimization is performed for a better tuning of the initial structure. If an appropriate structure is identified, the learning task can be accomplished by any suitable training algorithm such as the classical backpropagation algorithm [4], [5], [7], [8].

V. Experiments and Results





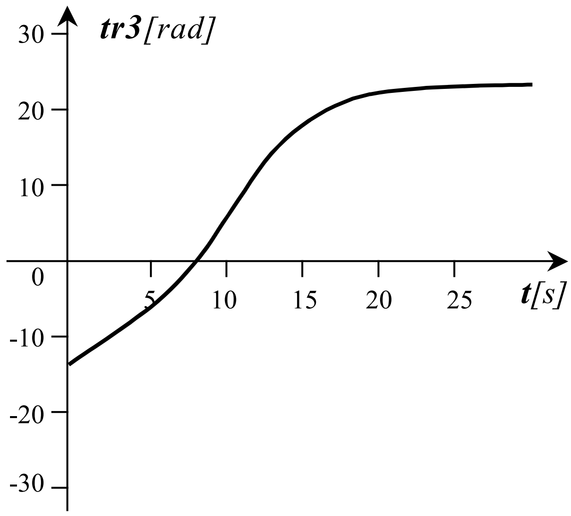
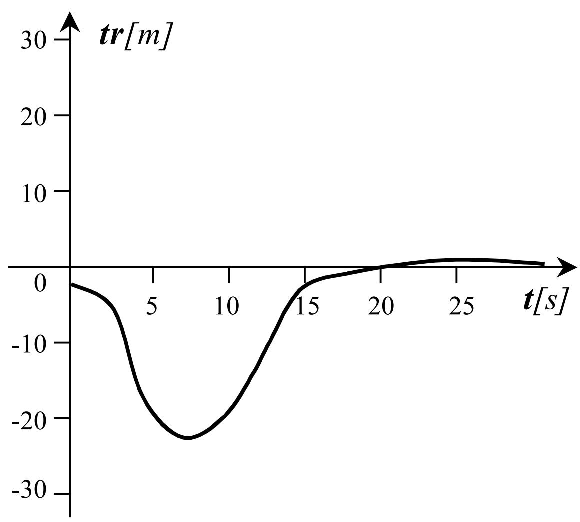


Fig. 3. Examples of Nominal Trajectories *tr1, tr2, tr3* (t[s] = discrete time)

The results of the simulation are illustrated on Fig. 3, 4 that display position tracking error. It can be seen that after the initial oscillation that is the result of system dynamics (inertial forces) swing of the local decays quickly as approaches to the final state, resulting in almost zero oscillations at the final time of the transfer.

Taken into consideration that the control algorithm is not too demanding in terms of processing time needed, proposed control scheme could be implemented on a real equipment.



ISBN: 978-988-17012-5-1

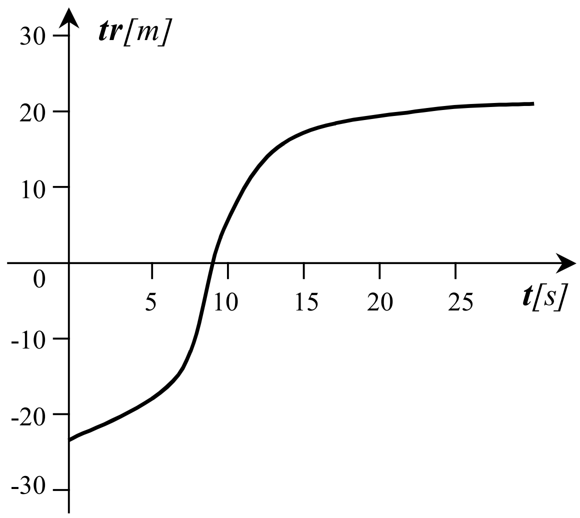


Fig. 4. Examples of Achieved Trajectories *tr* (t[s] = discrete time)

VI. Cutting Forces Optimization Via Genetic Algorithm

Stable and unstable cutting experiments have been designed. As the magnitude and direction of the cutting force significantly influence the machining accuracy therefore their precise knowledge is required in precision finishing. There are process requirements concerning to the accuracy and quality that are continuously increasing [4], [10] - [13]. A genetic algorithm approach was applied to the simulation model to determine the parameter values of process that would result the simulated cutting forces in ball-milling.

We realize influence search of cutting speed on total average cutting force and its components at machining the various steel materials and then we realize the process optimization within genetic algorithm environment. The aim is to determine and optimize first of all a relation for all three components of the cutting force, which describes the reality well and can be applied relatively easily in the practice. Total average cutting force is calculated by following way:

$$F_c = \sqrt{F_x^2 + F_y^2 + F_z^2} \tag{11}$$

The cutting forces are depending essentially on the properties of material of used workpiece. The total average cutting force decreases when increasing cutting speed at machining. Cutting direction and cutting depth have no influence on the decrease of cutting forces when increasing cutting speed. The system for cutting process simulation contains the experimental model and the simulation model based on genetic algorithms.

Table I: Relevant Parameters

Number of generations	186
Probability of the reproduction	0.79
Probability of the selection	0.67
Probability of the mutation	0.0012
Elitism function	19
Period of regeneration	10

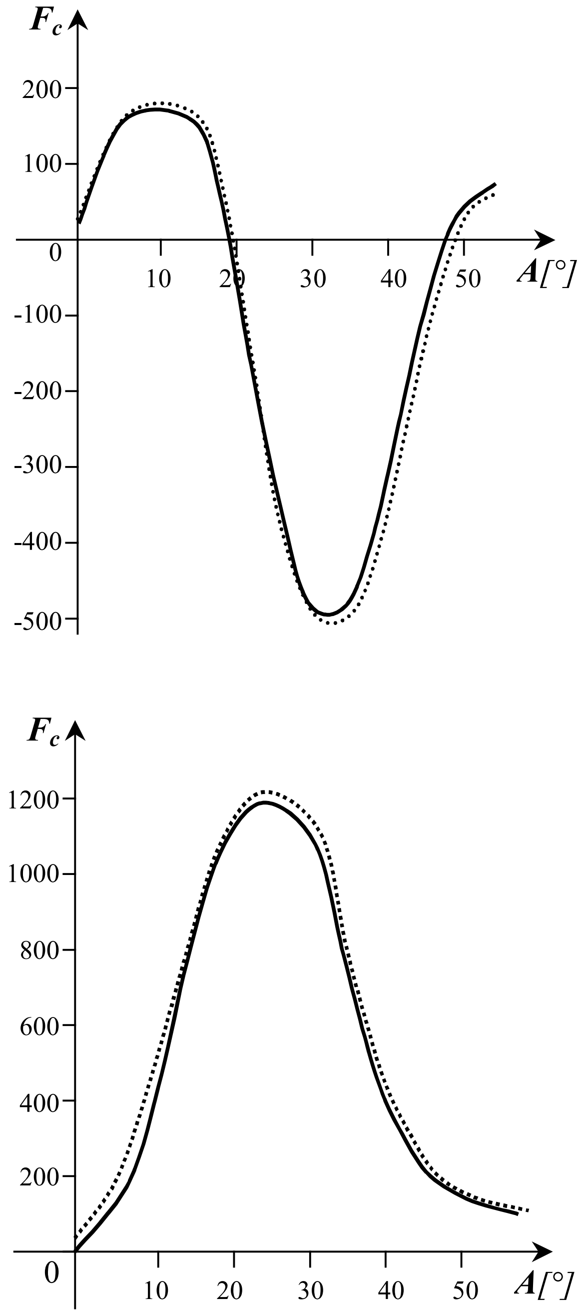


Fig. 5. Simulation and Measurement of Cutting Forces

Experiments were realized on the different materials of workpieces and cutting speeds. There is an example of the simulation model to determine the process parameter values via genetic algorithm approach. The simulation model would result the simulated cutting forces in ball-milling process. Results of realized experiments and the simulation processes are illustrated in Fig. 5., Table I. We compare the simulation and experimental results. Parameter F_c represents cutting force, parameter *A* represents angle of the cutter rotation. The dashed line represents the simulated cutting forces. The continuous line represents the experimental cutting forces. The model of cutting forces simulation is confirmed by experimental results.

Fitness evaluation course for used methodology for solved ball-milling process is shown in Fig.6.

ISBN: 978-988-17012-5-1

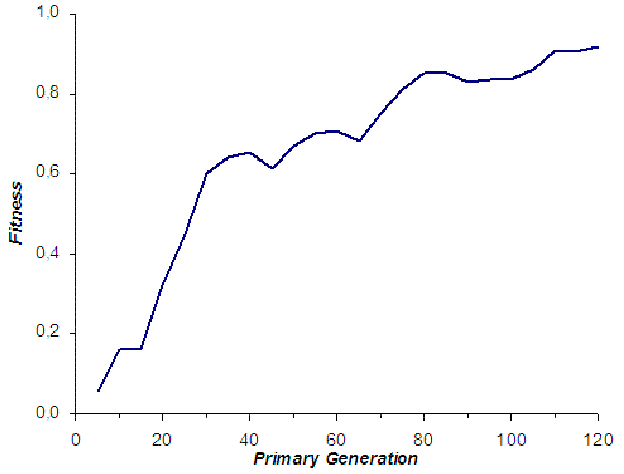


Fig. 6. Fitness Evaluation Course for Used Methodology Approach

VII. Conclusion

The system for cutting process simulation contains the experimental model and the simulation model based on genetic algorithms. The operation of the cutting forces simulation model can be confirmed by experimental results (see Fig.5).

Cutting forces are important parameters to predict machining performances of any machining operation. The force components can be obtained and displayed simultaneously on the screen for analyzing force changes via expert system tool. In the A/D board, the analogue signal will be transformed into a digital signal form. A connecting plan blocks and channel A/D interface board are realize within interface hardware module.

The predictive modeling of machining operations requires detailed prediction of the boundary conditions for stable, safety and reliable machining. Cutting forces are relevant factors to predict machining performances of any machining operation.

The classification of solved signal features for process condition monitoring has been carried out using fuzzy parameters optimization processing via genetic algorithm optimization methodology. We realize case adaptation by adjusting the fuzzy sets parameters. Fuzzy parameters within optimization procedure could be multiobjective. This genetic algorithm approach is suitable for fuzzy implementation control and for solution of large-scale problems. 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. This genetic algorithm is feasible in a multi-transputer environment.

Fuzzy rules based system with an optimization by genetic algorithms approach can be effectively used to obtain relevant results in real diagnostic system problem solving. The fuzzy theory may be combined with the genetic model, for instance by putting a value between „0“ and „1“ in the Boolean cluster code to act as object belonging to probability.

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. Nowadays, some achieved results seem to be very interesting. Future research will be also concerned on two populations approach only that occasionally communicate each other and on asynchronous version of co-evolution model that allows each population to have a slower and more stable evolution pace but also is suited for a parallel implementation in which each population is evolved on a separate processor. Such parallel implementation is important for the efficient processing of a large number of well logs simultaneously.

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. (Grant Number: VEGA 1/4133/07).

References

[1] Herrera-Viedma, E.: *Modelling the retrieval process of an information retrieval system using an ordinal linguistic approach.* American Society for Inf. Sc. 6/2001, pp. 460-475.
[2] Gallova, S.: *Fault Diagnosis of Manufacturing Processes via Genetic Algorithm Approach,* IAENG Engineering Letters, 2007, 15, Issue 2, 349-355.
[3] Rochio, I. J.: *Relevance feedback of Information Retrieval.* The Smart System Experiments in Automatic Document of Processing. Prentice-Hall, pp. 313-323, 1971.
[4] Gallova, S.: *A maximum Entropy Inference within Uncertain Information Reasoning,* in Proc. of Information Processing and Management of Uncertainty in Knowledge-based Systems, Les Cordeliers, Paris, 2006, 1803-1810.
[5] Ballé, P.: *Fuzzy Model-based Parity Equations for Fault Isolation.* Control Engineering Practice, No7, 1999, pp.261-270.
[6] Brini, A.: *Introduction de la Gradulaite dans le Jugement Utilisateur.* Dea Report. Toulouse, France, 2002.
[7] Pasi, G.: *A logical formulation of the Boolean model and weighted Boolean models.* Lumis'99, University College London, England, 1999
[8] Zadeh, L.: *The Concept of Linguistic Variable and It´s Application to Approximate Decision Making. Moscow.* Mir, 1976.
[9] Goldberg, D. E.: *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, Reading, Massachusets, 1989.
[10] Lu, M., Dong, F., Fotouhi, F.: *The Semantic Web. Opportunities and Challenges for next Generation Web Applications.* Information Research (7), 2002.
[11] Kruschwitz, U.: *An Adaptable Search System for Collections of Partially Structured Documents.* „Intelligent Systems, IEEE vol.18, pp. 44-52, 2003.
[12] Novakovic, B.: *Fuzzy logic control Synthesis without any rule base.* IEEE Trans. on Systems. Man and Cybernetics. Vol.29, No3 pp.459-466, 1999
[13] Rocacher, D. Bosc, P.: *About the set of Fuzzy relative Integers and the Definition of Fuzzy Bags on Zf.* IFSA, Springer – Verlag (2715), pp. 95-102, 2003.