16th IMEKO TC10 Conference
"Testing, Diagnostics & Inspection as a comprehensive value chain for Quality & Safety
Berlin, Germany, on September 3-4, 2019

Application of the fusion of regression machines for the analog circuit state identification

Piotr Bilski[1]

[1]Institue of Radioelectronics and Multimedia Technology, Warsaw University of Technology, Warsaw, Poland, pbilski@ire.pw.edu.pl, +48 601056969

***Abstract* – The paper presents the application of the combined group of regression algorithms to identify state of the analog circuit. Implementing the fusion of regression machines is aimed at obtaining high accuracy of the diagnosed system's state, especially compared to the single parameter identification algorithm. The large number of simple methods (such as linear regression techniques) is expected to give the high accuracy without the need of time consuming and complex optimization of the selected approach (such as Support Vector Machines – SVM). The approach consists in preparing the ensemble architecture, selecting computational methods, optimizing features extracted from the diagnosed system and testing the approach. The tests were conducted to evaluate efficiency of various fusion architectures, determine their accuracy for different sets of features and confront them against the single optimized regression algorithm. The time analysis verified the ability of the approach to use the framework in the online mode. Obtained results show the potential of the proposed framework for the accurate identification of analog system parameters, which can be used to analyse other types of systems.**

Keywords – regression, diagnostics of analog systems, parameter identification.

I. INTRODUCTION

The modern diagnostics of analog systems faces multiple problems, which are solved using various approaches, among which the most prominent belong to the Artificial Intelligence (AI) domain. One of the pressing issues is the accurate Parameter Identification (PI) in the system [1], which allows for determining its state during the long-term exploitation. Initially in the nominal state, the system degrades with time and its parameters change, starting from small deviations from the original values, further going beyond the tolerance regions. In the complex industrial systems and sophisticated devices it is important to learn in advance that they do not (or will not in the nearest future) operate exactly as they should. This way faulty elements may be replaced before the object becomes permanently damaged. Systems critical for society (such as power plants [2]) are constantly monitored, as their parameters must not deviate beyond the nominal state.

The PI is difficult because of multiple factors and phenomena. These include the noise present in the observed System Under Test (SUT) responses, tolerance regions, making the definition of the nominal state "fuzzy", existence of ambiguity groups [3], etc. Also, in most systems there is the non-linear relation between the parameters' values and the observed diagnostic features. These pose a challenge for the applied algorithm, usually leading to a long and mundane process of adjusting parameters. In many applications linear regression (approximation) approaches provide the acceptable accuracy. They can be the alternative to more complex methods, such as SVM [4].

The paper presents the fusion (or ensemble) architecture of AI-originated regression machines determining the state of the electronic circuit. Application of multiple algorithms estimating its parameters is confronted against the single, more sophisticated approach, i.e. .SVM regressor. It is assumed that voting mechanism allows multiple approximators to obtain the greater accuracy. The tested circuit contains discrete elements, which change over time and influence its work regime. Though integrated circuits are currently majority and in the case of leaving the nominal state they can be easily disposed of, in some applications, such as expensive audio amplifiers it is still justified to replace the damaged elements instead of exchanging it into the new device.

The paper structure is as follows. In Section II the parameter identification task is introduced. Section III discusses the proposed fusion architecture with the characteristics of the usable algorithms. In Section IV details of the prepared solution (such as voting mechanism) are considered. Section V presents the SUT, while in Section VI conducted tests are presented. Section VII contains conclusions and future prospects.

II. AI-BASED MULTIPLE PARAMETER IDENTIFICATION PROBLEM

Every SUT, disregarding its technical nature (electronic circuits, electrical machines, automation elements, etc.), should operate according to the design principles, defining the nominal state. It depends on the configuration of SUT parameters $\boldsymbol{p}=\{p_1, ..., p_m\}$ (such as capacitances, resistances' values, amplification coefficient, etc.), which determine produced output (response) signals $\boldsymbol{y(t)}$ as reaction to excitations $\boldsymbol{x(t)}$.

$$\boldsymbol{y}(t) = f(\boldsymbol{x}(t), \boldsymbol{p}) \tag{1}$$

The task is to construct the function *g()*, which will be able to map information represented by SUT responses into the approximated set of parameters $\widehat{\boldsymbol{p}}$, which should be as close to the actual set $\boldsymbol{p}$ as possible (usually in the least squares sense).

$$\widehat{\boldsymbol{p}} = g(\boldsymbol{y}(t), \boldsymbol{x}(t)) \tag{2}$$

Unfortunately, the function *f()* (1) is not easily reversible, therefore based on the observation of responses it is difficult to determine exact values of parameters, even if excitation patterns $\boldsymbol{x(t)}$ are known. The relation (1) is usually described by the computer model, therefore it is easy to establish the approximations $\widehat{\boldsymbol{p}}$ heuristically, by generating multiple simulations for varying configurations of SUT parameters. This allows for creating the diagnostic module, representing the function (2), i.e. the function *g()*.

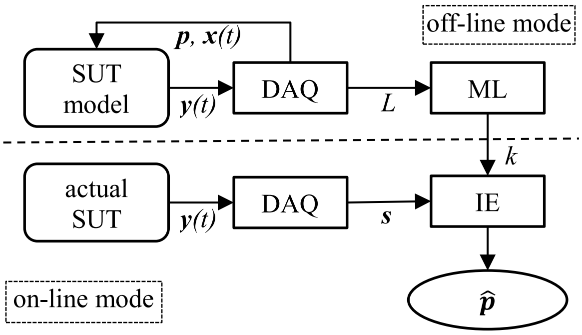

Fig. 1. Architecture of the PI module

Currently the most popular are data-driven approaches. They enable defining the PI task based on the algorithm extracting knowledge from available sets, for instance through the machine learning process [5]. The sets are first created during the simulations of different SUT states. The diagnostic features (symptoms) are extracted from the output signals to form the symptoms' vectors $\boldsymbol{s}$, based on which knowledge k about the actual SUT state can be induced. The AI-based identification implements the scheme in Fig. 1, operating in the off-line and on-line mode. In the former the DAQ module initiates model simulations, acquires response signals and extracts symptoms from them. The Machine Learning (ML) module operates on the learning set L to generate knowledge k connecting observed symptoms and predicted parameters' values. It is the core of the Inference Engine (IE), which (working in the on-line mode) will process the vector of symptoms extracted from the actual SUT and estimate values of parameters $\widehat{\boldsymbol{p}}$.

Contrary to the popular fault detection and location scheme, where the system must be classified as working correctly or damaged (described by the discrete category), in the presented PI task the expected outcome is the predicted real-number value of the parameter p_i. According to Fig. 1, it can be described as follows:

$$\{k(L), \boldsymbol{s}\} \rightarrow \mathbb{R}^m \tag{3}$$

Data sets used in AI-based identification consist of vectors of diagnostic symptoms $\boldsymbol{s}_i$ (further also called examples) coming from model simulations for the specific configurations of parameters $\boldsymbol{p}$, which values describe each example. This way it is possible to perform supervised learning, discovering relation between symptoms and SUT parameters. The scheme requires two data sets of the same form (4), used for training and testing (L and T, respectively) the regression machine. From L knowledge k is extracted, while T is used to test generalization abilities of the regression method. Both sets shoudl then contain different, exclusive examples.

$$L \equiv T = \begin{bmatrix} \boldsymbol{s}_1 & \boldsymbol{p}_1 \\ \vdots & \vdots \\ \boldsymbol{s}_n & \boldsymbol{p}_n \end{bmatrix} = \begin{bmatrix} s_{11} & \cdots & s_{1l} & p_{11} & \cdots & p_{1m} \\ & \ddots & & & \ddots & \\ s_{n1} & \cdots & s_{nl} & p_{n1} & \cdots & p_{nm} \end{bmatrix} \tag{4}$$

Quality of generated knowledge is measured using one of the possible determination scores. In the presented research the measure (5) is used [6], where p_{ij} is the actual value of the *i-th* SUT parameter for the *j-th* example from the set, $\hat{p}_{ij}$ is its estimated value and $\bar{p}_{ij}$ is the average of the actual values. The maximum possible score is equal to 1 (when the predicted and actual parameters' values are identical) with the unlimited minimum going towards -∞. The score is calculated for each parameter separately. This way it is possible to optimize the regression machines regarding all parameters for the whole available set, by maximizing value of (5). In practice, only the range (0.5; 1) is useful (i.e. close enough to the actual values) PI.

$$acc(p_i) = 1 - \frac{\sum_{j=1}^{n}(p_{ij} - \hat{p}_{ij})^2}{\sum_{j=1}^{n}(p_{ij} - \bar{p}_{ij})^2} \tag{5}$$

The score (5) is used in the presented research to test the regression quality on L and T. Result for the former is used to configure the ensemble, while the latter shows the performance of the designed approach. Vectors of symptoms prepared for both sets should cover the most typical SUT states: nominal and the most probable deviations. It is assumed that the framework detects signle faults, i.e. only one parameter is beyond nominal, while all other are within their tollerance margins. Though the proposed scheme is able to identify values of all parameters indopendently, its efficiency for the multiple faults prediction must be evalutaed separately.

In the following sections the architecture of the classifiers' fusion is introduced. Next, the implementation of the ensemble as the software module is presented. To create the architecture, Python language was used, including the sci-kit package [7], which is the standard library of the AI-based algorithms.

III. THE PROPOSED FUSION OF REGRESSION ALGORITMS

To solve the PI problem, the single regression method is usually applied (such as RBF Artificial Neural Networks or SVM). These approaches consider non-linear relations between the parameters' values and observed symptoms, additionally SVM is able to cope with the data inconsistency. The disadvantage of these approaches is the need to tune them to obtain the best results. Alternatively, simpler, linear regression methods may be used here. Usually not as effective as their non-linear counterparts, they can be exploited in the more accurate way by combining them into the single regression machine.

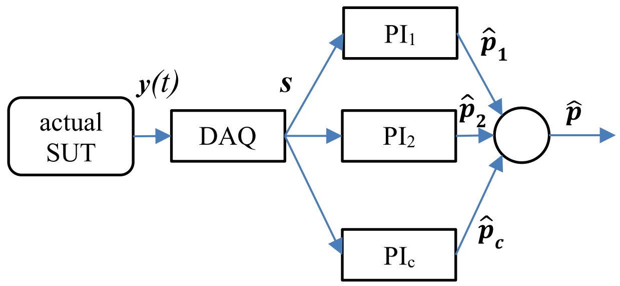

Fig. 2. Ensemble of regression machines

A. Architecture overview

Each algorithm in the ensemble is trained separately on the same set L (4), but knowledge k extracted from it depends on the particular method. Therefore it is expected that regression results will be different for each algorithm and the combined outcome can be better than of any other separate approach. The scheme in Fig. 2 is similar to the classifier ensemble, used in diagnostics before [8]. The difference lies in the way of calculating the output value, strongly related to the applied voting mechanism (which is investigated further).

The software architecture is aimed at training all approaches subsequently in the off-line mode on the previously generated set L. The multiple PI is performed by all algorithms working in parallel, the overall result is calculated after their job is complete.

The linear regression implements the following relation between the *i-th* SUT parameter and analysed symptoms [7]:

$$p_i = \sum_{j=1}^{l} s_j \cdot w_j + w_0 \tag{6}$$

where weights w_j are adjusted during the training process. Each algorithm introduces modifications to (6), leading to different PI results.

B. Considered regression algorithms

Among multiple algorithms the following were selected, based on their work regime and potential usefulness for the task. The simpler ones were preferred, to check their applicability for complex data processing:

- Least Squares (LS) regression, which implements directly equation (6) and provides the simple approximation of independent features.
- Ridge (R) regression, which modifies (7) by adding penalty for too large coefficients, imposing as simple approximation as possible.
- Bayesian Ridge (BR) regression includes the regularization coefficients to the approximation. The produces value of ***y(t)*** is treated as the random variable with normal distribution.
- Orthogonal Matching Pursuit (OMP) is the implementation of the linear regression by defining the specific number of non-zero coefficients w_i.
- Lasso Least Angle Regression with Information Criterion selection (LLIC) is the combined approach of high-dimensional data processing with sparse coefficients in (6) and the model selection using the Akaike or Bayesian criterion.
- Automatic Relevance Determination (ARD) is the variant of the Bayesian Ridge regression with the sparse coefficients w_i.
- Theil-Sen (TS) estimation is the non-parametric method insensitive to outliers, with the result curve calculated as the median among data points.

Other possible algorithms, including Passive Aggressive estimator or Elastic Net were initially tested and discarded due to their low scores. Each algorithm has multiple copies, responsible for their individual SUT parameter. The *i-th* PI machine in Fig. 2 consists of *m* separate approximators, generating together the vector of values $\hat{\boldsymbol{p}}$. If needed, the fusion may be supplemented by other algorithms.

The presented approach was confronted with the more sophisticated approximation algorithm, i.e. SVM regressor. It is a well-established approach, especially for processing uncertain data. Its disadvantage is the time-consuming process of selecting optimal parameters of the kernel. The aim of the experiment was to check if the ensemble can outperform the SVM-based approximator.

IV. ARCHITECTURE DETAILS

During the ensemble construction, it is crucial to determine the set of algorithms and their significance during the overall result calculation. Initially, all available methods were tested on the provided learning data sets, then *c* the best ones were selected for the fusion, which was finally evaluated on the testing sets. The comparison was made for each parameter separately. It turned out that there is the subset of regression methods being more applicable to the PI task than others, but it is not possible to point out the single one dominating all of them. This justifies using the ensemble rather than the single method. In this section issues related with the architecture construction are presented. For each parameter the ensemble is constructed only of algorithms with the score on L high enough.

A. Voting mechanism

To generate the single output out of the intermediate values produced by different approximators, they must be combined in a way reflecting significance of each algorithm. The proposed relation for the *i-th* parameter estimation $\hat{p}_i$ is as follows:

$$\hat{p}_i = \frac{1}{c} \cdot \sum_{j=1}^{c} \hat{p}_{ij} \cdot \alpha_i \tag{7}$$

where $\boldsymbol{\alpha}=\{\alpha_1, \ldots, \alpha_c\}$ is the vector of weighting coefficients for subsequent regression methods, calculated based on the performance on L. This way the most accurate approaches influence the most the estimation result. The weights are calculated as follows, where $acc_j(p_i)$ is the score of the *j-th* algorithm of the *i-th* estimated SUT parameter:

$$\alpha_j = \frac{\frac{1}{1-acc_j(p_i)}}{\sum_{i=1}^{c}\frac{1}{1-acc_j(p_i)}}, \quad \sum_{j=1}^{c}\alpha_j = 1 \tag{8}$$

All weights are scaled to represent values between *0* and 1 and they are proportional to the accuracy (5). To avoid influencing the overall result by the regressor with the low score on the training set, the weight if calculated provided that $acc_j(p_i)$ is greater than the threshold θ, which was set to *0.45*. When the score is lower, the algorithm does not take part in voting (its weight is zero).

B. Features significance

The estimation accuracy depends on the symptoms available to the algorithm. Without the prior knowledge about the features importance, all should be used. Their large number may significantly degrade regression. Therefore elimination of redundant and not-important features was proposed. Besides the correlation analysis (to find dependent variables), there is the need to determine the set of symptoms representing changes in the specific parameter. This way it could be easier to decompose PI task for different parameters into independent routines, based on different sets of symptoms. Due to the existence ambiguity groups, such a separation may not be always possible. The algorithm is presented in Fig. 3.

features_selection(L)
    foreach p_j
        $L(p_j) \leftarrow \emptyset$
        foreach s_i
            if change(s_i, p_j) == TRUE
                $L(p_j) \leftarrow L(p_j) \cup s_i$

Fig. 3. Features selection algorithm

For each SUT parameter the subset of features from the learning set L is identified, depending on whether their change determines the change of the corresponding symptom's value. The features' values should be beyond the 10% margin from their values in the nominal state (the **change**() function). In result the regression machine estimating the selected parameter operates on the individual set of symptoms (represented by $L(p_j)$). The approximators' scores for the original and reduced sets were compared to verify which gives better results.

V. EXPERIMENTAL TEST STAND

The proposed architecture was tested on the 5th order lowpass analog filter, presented in Fig. 4. Its structure contains three operational amplifiers, five capacitors and resistances. The latter two groups of elements were subject to PI task, based on the symptoms extracted from the SUT responses. These are output signals generated after providing the sinusoidal excitation (1V amplitude and 9kHz frequency, i.e. close to the 10kHz cut-off frequency of the circuit) to the SUT input (node No. 1). Responses were recorded at accessible nodes No. 2, 3, 5, 6, 8 and 9. From output patterns the first three maxima, minima, zero crossings and their corresponding time instants were extracted as symptoms. This way each symptoms vector contains 54 features, which can further be minimized.

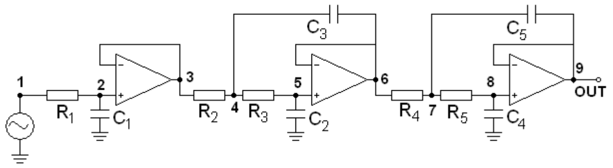

Fig. 4. Fifth order lowpass filter

The nominal values of identified elements are as follows: $R_1=R_2=R_3=R_4=R_5=1k\Omega$, $C_1=16nF$, $C_2=19nF$, $C_3=13nF$, $C_4=51nF$ and $C_5=49nF$. The model of the SUT was implemented in the Simulink environment. Experiments consisted in setting the selected parameter out of tollerance margins, while leaving all remaining ones within the nominal state (affected by the random variable representing tollerance dispersion and the uniform noise). Elements' changes ranged from 10% to 90% of the nominal value (in both directions, i.e. above and below it).

VI. EXPERIMENTAL RESULTS

The experiments were divided into a couple of stages, described in the subsequent sections. Firstly, all described algorithms were tested separately to check their usefulness to estimate SUT parameters. Next, algorithms selected to ensemble were combined to form the PI module and confronted against SVM regression. Finally, the features' selection algorithm was implemented to increase the regression accuracy if possible.

Four pairs of data sets (i.e. L and T) were used for tests. The first two (L_1, T_1 and L_2, T_2, respectively) included 70 and 180 examples, with actual parameters' values not affected by tolerances. The remaining two pairs (L_3, T_3 and L_4, T_4), also containing 70 and 180 examples respectively, included simulations affected by elements' tolerances and noise. The smaller sets covered 7 simulations for each diagnosed element, while the more extensive tests included 18 changes of each parameter's value.

A. Accuracy of selected regression algorithms

Algorithms introduced in Section III.B were tested separately on full vectors of symptoms. Results on the set L_1 for each parameter are in Table 1, while for the set T_1 are in Table 2. The bold font indicates the highest score, while italics show, which algorithm's accuracy allows for including it in the ensemble. The "0.00" values indicate the approximator was unable to converge on the learning set, therefore it will not take part in voting. In most cases the LS regression is the best on the learning set, but as Table 2 shows, knowledge extracted in this case is not well generalized, which justifies application of the fusion. Although the analysis of L_1 is dominated by two methods: LS and TSR, knowledge verification on T_1 shows more algorithms involved in the accurate regression: also BR, LLIC have the highest scores for some parameters. Table 2 also shows, which SUT elements are difficult to identify: for R_1, C_1 and C_2 it is hard to get scores high enough to make them usable.

Table 1. Regression scores on the set L_1.

	LS	R	BR	OMP	LLIC	ARD	TSR
R_1	**0.95**	0.41	*0.49*	*0.50*	*0.50*	*0.50*	0.09
R_2	*0.91*	0.18	*0.67*	*0.70*	*0.80*	*0.68*	**0.98**
R_3	**0.97**	0.10	*0.73*	*0.72*	*0.75*	*0.70*	0.89
R_4	**1.00**	0.26	*0.73*	*0.66*	*0.71*	*0.74*	0.99
R_5	**1.00**	0.08	0.00	*0.64*	*0.65*	0.38	0.97
C_1	**0.95**	0.41	0.48	0.00	0.00	0.10	-0.87
C_2	*0.92*	0.04	0.10	0.00	0.00	0.00	**0.98**
C_3	*0.90*	*0.52*	*0.65*	0.00	0.00	0.01	**0.97**
C_4	**1.00**	0.07	0.11	*0.55*	0.00	0.00	0.99
C_5	**1.00**	0.27	0.40	*0.53*	0.00	0.00	0.96

Results on other data sets are similar, still showing advantage of LS and TSR over other approaches, but after introducing tolerances and noise the maximum scores are dispersed among other methods. Symptoms affected by these phenomena are more difficult to process, therefore regression accuracy for T_3 and T_4 is lower (on average by *0.1*) than for T_1 and T_2. Increasing the number of examples in the set did not improve scores. Accuracy on the testing set is always lower on the learning one, which suggests the loss of generalization during the training.

Table 2. Regression scores on the set T_1.

	LS	R	BR	OMP	LLIC	ARD	TSR
R_1	-0.09	0.41	0.49	0.50	**0.51**	0.50	-2.20
R_2	-1.21	0.18	0.68	0.68	**0.82**	0.68	0.42
R_3	-1.23	0.10	0.73	0.72	**0.74**	0.69	0.29
R_4	0.93	0.25	0.74	0.64	0.71	0.75	**0.94**
R_5	**0.90**	0.08	0.00	0.61	0.66	0.35	0.83
C_1	-0.01	0.41	**0.47**	0.00	0.00	0.10	-1.59
C_2	-1.24	0.04	**0.12**	0.00	0.00	0.00	-0.02
C_3	-1.16	0.51	**0.64**	0.00	0.00	0.01	0.44
C_4	0.92	0.06	0.10	0.53	0.00	0.00	**0.93**
C_5	**0.92**	0.31	0.44	0.50	0.00	0.00	0.77

B. Efficiency of the ensemble

Based on the information from algorithms used individually, the ensemble was constructed with the voting mechanism defined by (7) and (8). In Table 3 results for the test T_3 are presented. In the subsequent columns, scores of the algorithms (with least one the most accurate estimation) treated separately, ensemble (ENS) and the SVM regressor (SVR) outcomes are shown. The latter was designed to use the RBF kernel with the optimal width $\sigma=0.001$. Again, bold font indicates the best estimation. In all cases the fusion is more accurate than SVM, which requires complex and time-consuming tuning (covering selection the best kernel and adjusting its parameters). The overall performance shows that although not the best for all parameters, the fusion of regressors is the most reliable approach, maintaining high score for all parameters.

Analysis of the ensemble performance has also shown that in some cases eliminating the LS from the fusion leads to increasing the score, as this method fails to properly generalize knowledge about values of R_1, C_1 and C_2.

Table 3. Regression scores on the set T_3, including ensemble and reference method.

	LS	BR	LLIC	ENS	SVR
R_1	**0,63**	0,49	0,50	0,59	0,34
R_2	0,81	0,69	0,66	**0,82**	0,27
R_3	0,77	0,66	0,71	**0,78**	0,65
R_4	0,35	0,70	**0,74**	0,45	0,38
R_5	0,56	**0,79**	0,69	0,70	-0,17
C_1	**0,68**	0,50	0,00	0,55	-0,02
C_2	0,82	0,23	0,00	**0,83**	0,63
C_3	**0,77**	0,69	0,00	0,75	0,48
C_4	**0,81**	0,13	0,00	0,80	0,23
C_5	0,37	0,55	0,00	**0,59**	-0,11

The actual accuracy of regression is presented in Fig. 5. Here variability of the selected parameter, i.e. R_5 for the set T_4 (180 examples with tolerances and noise) with its estimated values obtained by the LS regression and fusion of all methods from Section III.B are shown. Both outcomes are close to the actual values, especially around the nominal range, eliminating the threat of the false alarm.

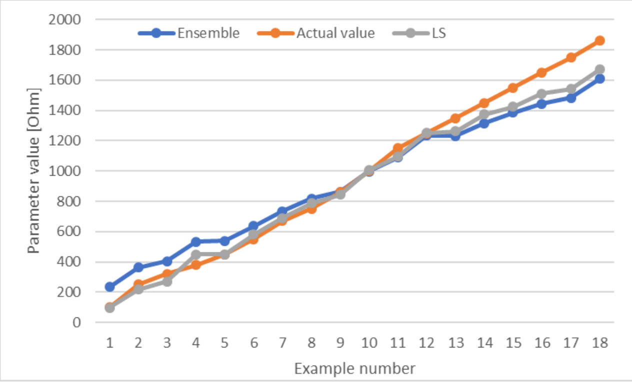

Fig. 5. Comparison between the regression accuracy of the Least Squares and ensemble regression

C. Influence of the feature selection algorithm

Experiments presented so far were conducted in the full data sets, containing all 54 symptoms, extracted from the SUT responses at accessible nodes. Introduction of the algorithm from Section IV.B was aimed at decreasing the amount of data processed by the algorithms, and hopefully maximizing the approximation scores. For each SUT element a different subset of symptoms was returned, causing different conditions for training the regression algorithms. In Table 4 results for processing the set T_1 by all algorithms are presented, including their ensemble. Outcomes for the symptoms' reduction method are inconclusive, as in some cases (C_2, C_3, C_4) the performance of both individual algorithms and ensemble has increased, but in other the accuracy dropped (especially for R_5, which is caused by the elimination of too many symptoms).

Table 4. Regression scores on the set T_1 for the reduced sets of symptoms.

	LS	R	BR	OMP	LLIC	TSR	ENS
R_1	-0,04	0,19	0,26	**0,49**	0,48	0,00	0,32
R_2	0,41	0,12	0,41	0,23	**0,41**	0,00	0,38
R_3	**0,68**	0,00	0,00	0,45	0,56	0,00	0,64
R_4	**0,33**	0,11	0,30	0,30	0,32	0,00	0,28
R_5	0,15	0,05	0,10	**0,16**	0,15	0,00	0,12
C_1	**0,75**	0,39	0,49	0,00	0,00	-31,28	0,47
C_2	**0,70**	0,02	0,10	0,00	0,00	0,13	0,36
C_3	**0,85**	0,32	0,45	0,00	0,00	0,33	0,57
C_4	0,92	0,06	0,10	0,53	0,00	0,91	**0,93**
C_5	0,92	0,31	0,44	0,50	0,00	0,85	**0,93**

VII. CONCLUSIONS

The proposed ensemble of regression machines has proven the usefulness during the estimation of parameters of the electronic analog circuit. Although it is a complex system with multiple parameters to identify simultaneously, most parameters can be evaluated with the acceptable accuracy despite the noisy conditions and influence of elements' tolerances. In most cases simple linear dependencies between the symptoms and actual SUT parameter values are accurate enough to use simple methods, not requiring the extensive tuning.

The ensemble of regressors proven to be useful for the PI, as the combined operation of multiple algorithms gives in the global term better accuracy than any method used individually. The criterion of selecting the approach to the fusion based on its performance on the learning set is effective, but in some cases knowledge is not well generalized, which can be supressed by other approaches.

The features selection algorithm requires modifications and extensions, as its usefulness is currently limited. Accuracy for some parameters is improved, while for other – degraded. The most probable solution for this problem is the introduction of additional criteria for leaving or discarding the specific feature. Also, additional tests of ensemble with additional methods, not tested here (such as CART), should be performed in the future.

REFERENCES

[1] **Moradi, M.; Naraghi, M.; Nikoobin A.**: Parameter identification of nonlinear systems using indirect solution of optimal control problem, *Proceeding of the 3rd RSI International Conference on Robotics and Mechatronics*, Oct. 7-9 2015, Tehran, Iran, DOI: 10.1109/ICRoM.2015.7367797.

[2] **Lemma T.A.**: A Hybrid Approach for Power Plant Fault Diagnostics, *Studies in Computational Intelligence*, Springer, 2018,.

[3] **Stenbakken G.N.; Souders T.M.;. Stewart G.W.**: Ambiguity groups and testability, *IEEE Transactions on Instumentation and. Measurement*, Vol. 38., No. 5., 1989, pp. 941-947, DOI: 10.1109/19.39034.

[4] **Ravikumar B.; Thukaram D.; Khincha H.P.**: Application of support vector machines for fault diagnosis in power transmission system, *IET Generation Transmission & Distribution*, Vol. 2., No. 1., 2008, pp. 119-130, DOI: 10.1049/iet-gtd:20070071.

[5] **Mitchell T.**: Machine learning, *McGraw-Hill*, 1997.

[6] **Devore J. L.**: Probability and Statistics for Engineering and the Sciences (8th ed.), Boston, MA: Cengage Learning, 2011, ISBN 978-0-538-73352-6.

[7] **Pedregosa F.; et al.**: Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research*, Vol. 12., 2011 pp. 2825-2830.

[8] **Bilski P.**: Analysis of the classifier fusion efficiency in the diagnostics of the accelerometer, *Measurement*, Vol. 67., 2015, pp. 116–125.