

# DESIGN OF AN APPLICATION FOR AN EXPERIMENTAL IDENTIFICATION OF A SYSTEM IN MATLAB/SIMULINK ENVIRONMENT

Ladislav NYULÁSZI, Peter BUTKA, Vladimír GAŠPAR

Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice, Slovak Republic, Tel.: +421 55 602 4214,  
E-mail: ladislav.nyulaszi@tuke.sk, peter.butka@tuke.sk, vladimir.gaspar@tuke.sk

## ABSTRACT

*This paper focuses on creating an application in the MATLAB/Simulink environment for experimental identification of a system. It theoretically characterizes the experimental identification of a system and the choice and classification of identification methods. It also describes chosen methods of experimental identification (deterministic methods, neural networks) in details. A proposal of an application consisting of a user interface and a generated model scheme are also mentioned. The testing results on data from a small turbojet engine MPM-20 are shown. Based on the comparison of real measured values and output values of the model the application evaluates the accuracy of each of the identification methods. The main contribution of the proposed application is an automatization and simplification of the experimental identification process in MATLAB.*

**Keywords:** experimental identification, graphical user interface, MATLAB, Simulink, neural network

## 1. INTRODUCTION

In the past, a lot of attention has been given to physical, natural and chemical laws, from which the mathematical models were derived by deduction. This process is called the analytical modeling. As the time past, the attention drifted to experimental methods. These experimental methods, however require the existence of a physical object and a possibility to carry out experimental tests. Experimental methods are also costly by means requirements on the aperture equipment. An inseparable part of designing of different complicated devices is the creation of mathematical models using computers.

Real object and its model belong to the general terms of the experimental system identification. The real object is understood as an original or a real device, on which it is possible to carry out measurements, observations, the aim of which is to get to know internal relation in the system [1], [2].

Modeling expresses the experimental process, during which we assign a physical or an abstract model to the modeled system based on chosen criteria. It is one of the oldest methods of obtaining knowledge that originated from mimicking natural phenomena up to modeling on the principal of geometrical similarity. Apart from a physical model, we can also assign the system an abstract model (mathematical model – MM).

Although, the mathematical model does not allow to carry out experiments with physical basis, it allows to research phenomena ongoing on the original system using their mathematical description [2], [3].

The paper focuses on designing an application for a system's experimental identification, which consists of a graphical user interface (GUI) and block schemes of the final model. The main aim is to simplify the process of creating experimental models with the possibility to compare methods of experimental identification based on their accuracy.

## 2. METHODS OF EXPERIMENTAL IDENTIFICATION

The experimental identification (EI) of systems addresses the investigation of dynamic properties of a system and stating its mathematical model using an experimental approach. Based on the testing signals we influence the system, record its response and subsequently state the model based on the measured signals [2].

Methods of experimental identification require the existence of the researched object and a possibility to carry out experiments on this object. However, they do not require total knowledge of the system's structure and description of its processes. In order to create experimental models, we have chosen five methods that belong to two groups: deterministic methods and artificial neural networks.

### 2.1. Deterministic Methods

#### 2.1.1. Method of Successive Integration

The method of successive integration (MSI) [3], [4] allows the approximation of the transition function of a system on other output signal in form of a step function. This means that the output signal brings the system from one balanced state into another balanced state. These states are not necessarily the same.

In the first step, from the input signal values  $u(t)$  and output signal values  $y(t)$  we identify the coefficients of a differential equation ( $a_0, a_1, \dots, a_n$ ), that represent a mathematical model of a system

$$a_n y^{(n)}(t) + a_{n-1} y^{(n-1)}(t) + \dots + a_1 \dot{y}(t) + a_0 y(t) = u(t) \quad (1)$$

From the ration of the values of the input and the output signal the coefficient  $a_0$  in the steady state is evaluated as follows

$$a_0 = \frac{u(\infty)}{y(\infty)}. \quad (2)$$

For evaluating the  $a_1$  and  $a_2$  coefficients it is necessary to get a transformed differential equation, which we can get by subtracting a general relation for steady state from the linear differential equation

$$a_n y^{(n)}(t) + a_{n-1} y^{(n-1)}(t) + \dots + a_1 \dot{y}(t) + a_0 [y(\infty) - y(t)] = -[u(\infty) - u(t)] \quad (3)$$

After integrations in constraints from 0 to infinity with the stated initial and terminal conditions

$$u(0) = u_0 = 0; u(\infty) = u_\infty \neq 0 \neq u0, \quad (4)$$

$$y(0) = y_0 = 0; y(\infty) = y_\infty \neq 0 \neq y0, \quad (5)$$

we get a coefficient  $a_1$  in form

$$a_1 = \frac{a_0 \int_0^\infty [y(\infty) - y(t)] dt - \int_0^\infty [u(\infty) - u(t)] dt}{y(\infty)}. \quad (6)$$

By continuing with the integrations we can obtain the following coefficients, although the more we continue, the more the calculation error increases over the allowed border. This means that the methods of successive integration are practically usable only for systems of 2nd degree at maximum.

### 2.1.2. Area Method

Area method (AM) [3], [4] has basis in a transfer function in the form:

$$G(s) = \frac{K}{a_n s^n + \dots + a_1 s + 1} = \frac{K}{1 + \sum_{i=1}^n a_i s^i} = \frac{K}{N(s)}. \quad (7)$$

With a ratio of settled input values and output it is possible to calculate the amplification K:

$$K = \frac{y(\infty)}{u(\infty)}. \quad (8)$$

For the unit step on the input side we get a transfer function  $h(t)$ . It holds that:

$$K = h(\infty). \quad (9)$$

This method starts from the consecutive calculation of unknown coefficients of the system's model using calculation of progressively defined areas.

Area of the first degree  $S_1$  is defined by the relation:

$$S_1 = \int_0^\infty [K - h(t)] dt. \quad (10)$$

Using the Laplace transformation, we can express the area  $S_1$  as:

$$S_1 = \lim_{s \rightarrow 0} s \cdot \frac{1}{s} \left[ \frac{K}{s} - \frac{K}{N(s)} \cdot \frac{1}{s} \right]. \quad (11)$$

In order to state the coefficients  $a_1, a_2, \dots, a_n$  of the researched model, it is necessary to evaluate the area of the  $i$ -th degree. Their evaluation using a  $i$ -fold integration is influenced by error. This error increases with each integration. Because of this reason, it is convenient to evaluate suitable areas  $S_i$  using a single integration and evaluating a so called momentum function of the  $i$ -th degree  $M_i$ .

$$M_i = \int_0^\infty \{K - h(t)\} \frac{(-t)^i}{i!} dt. \quad (12)$$

For the area  $S_i$  it holds:

$$S_i = \sum_{j=0}^{i-1} S_{i-1-j} M_j + M_{i-1}. \quad (13)$$

Coefficients of the transfer are evaluated of the  $i$ -th area using an expression:

$$a_i = \frac{S_i}{K_i}, \text{ pre } i = 1, 2, \dots, n. \quad (14)$$

More accurate model can be obtained with continuous evaluation of coefficients  $a_i$ . After obtaining sufficiently accurate model, or if the evaluated coefficient is negligibly small, the calculation terminates.

### 2.2. Artificial Neural Networks

A neural network (NN) [5], [6], [7] is a massively parallel processor with inclination to storing experimental knowledge and their later usage. Neural networks mimic the human brain by collecting knowledge during learning and their storing using interneuron connections. Neural networks consist of neurons (mutually connected elements) working in parallel. Neurons and NN were inspired by a biological nerve system. Similarly as in nature, their function is stated by connection between neurons. Based on the configuration of the connections (synaptic weights) and network connections with the external environment, it is possible to train such a network and configure its properties, so that it will carry out a specific set of functions.

#### 2.2.1. Time Delay Neural Network

Time Delay NN [8], [9] are feedforward neural networks that allow the multilayer network to use an "eye to the past". This means that except for the momentary

input (in time  $t$ ), they can also “see” the inputs from previous  $D$  time elements ( $v$  times  $t-1, t-2, \dots, t-D$ ). TDNN can be trained using a classic procedure of error back propagation, although it is possible to sustain the order of the training samples in the training set.

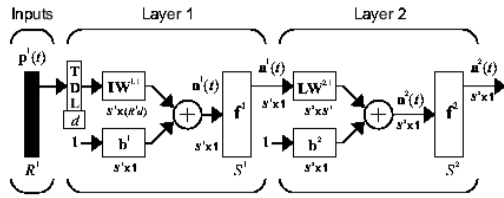


Fig. 1 TDNN architecture [10]

The scheme in Fig. 1 is suitable to predict time series. It contains a dynamic component in input and does not contain feedback. This is why the learning process is very fast.

2.2.2. Distributed Delay Neural Network

Distributed Delay NN [11] is a network type that distributes stored delay links throughout the entire network. The Fig. 2 shows a double layer DDNN with delay on both layers. A drawback of this network is a time consuming learning process because dynamic variants of backpropagation are using during learning.

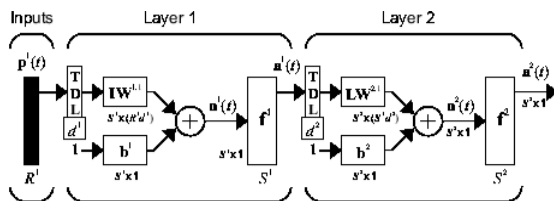


Fig. 2 DDNN scheme [10]

2.2.3. NARX Neural Network

NARX NN (nonlinear autoregressive network with exogenous inputs) [12] is a recurring dynamic network with feedbacks between various network layers. Model of this NN is based on a linear ARX model. To model nonlinear dynamic systems, in which the previous output is used as an input we use NARX networks. This approach is more beneficial, because the input of the feedforward NN is more accurate, by which it positively influences accuracy of the resulting network.

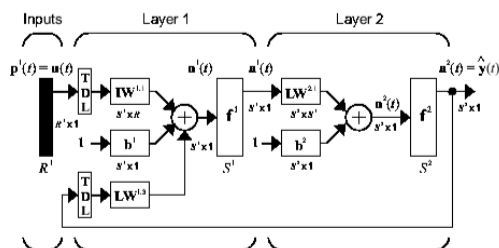


Fig. 3 NARX scheme [10]

3. DESIGN OF AN APPLICATION FOR EXPERIMENTAL IDENTIFICATION

The designed application for experimental identification of a system consists of two main parts. The graphical user interface (GUI) in the MATLAB environment and block schemes of the experimental models in Simulink environment. The GUI design can be seen in Fig. 4.

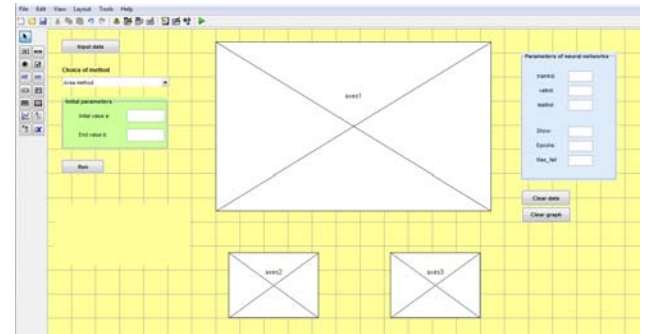


Fig. 4 Design of the user interface

The application contains the following functionalities:

- 1. Reading the input data** – The user opens the .mat file with data and selects data input and output.
- 2. Choice of method of experimental identification (EI)** – Possibility to choose from five methods. Two of them are deterministic (*Method of successive integration and Area method*). The other 3 types are artificial neural networks (*Time Delay Neural Network, Distributed Delay Neural Network, NARX Neural Network*).
- 3. Initial parameters** – Is it possible to enter a range of input data, that will be used while creating a model using methods of experimental identification. It is appropriate mostly in deterministic methods, where models are evaluated using a leap signal change. In neural networks, the values are automatically set according to the length of the input data.
- 4. Parameters of neural networks** – Manual entering of several parameters for training of neural networks. Training, testing and validation ratio can be set here (*trainInd, testInd, valInd*). Other settable parameters are *Max\_fail* (Maximum validation failures), *Epochs* (Maximum number of epochs to train) and *Show* (Epochs between displays).
- 5. Graphs** – Three windows for displaying graphs are viewable in the application. Two smaller are used to show input (axes2) and requested output (axes3). The larger graph (axes1) shows results of experimental identification in form of colored distinct measured output data and data that were obtained using the selected method.
- 6. Data and graphs cleansing** – It is also possible to delete resulting graphs with a single button click and continue creating the experimental models with read data and configuration. The second possibility is to entirely clean all data and read a new set.

- 7. **MAPE** – To set the model accuracy based on the difference between measured data and output model is evaluated as a mean absolute percentage error (MAPE).
- 8. **Error messages** – The application contains error messages that warn the user in case of entering incorrect values.

**3.1. Designing the block schemes of experimental models**

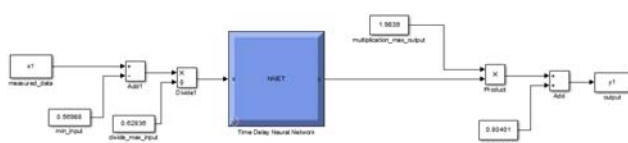
Block schemes can be created in two ways. The first is their creation using libraries that are implemented in the Simulink environment. The second way is creating the block schemes using commands in MATLAB. For the purpose of our solution, we have chosen the second alternative because it has the following advantages [4]:

- The user can create various other commands (evaluations, measurements, data preprocessing, ...) before the creation of the system. These commands can be automatically included into parameters in each block scheme. Before the model is drawn, the blocks or parameters can be altered based on before stated criteria.
- When changing a block in a scheme, it is enough to change its name in the code and launch the model redraw.
- Using commands, a complete draw of the scheme is faster than by moving the blocks from the library in the Simulink environment.

The resulting block schemes slightly differ for deterministic methods and neural networks. In case of the deterministic methods, the model of the system is entered in form of a transfer function in a discrete form into the *Discrete Transfer Fcn* block. An example of the scheme for the area method is shown in Fig. 5. In case of the neural networks, the evaluation model of the system is represented by a neural network block, generated by the *gensim* command. The scheme of a Time Delay Neural Network can be seen in Fig. 6.



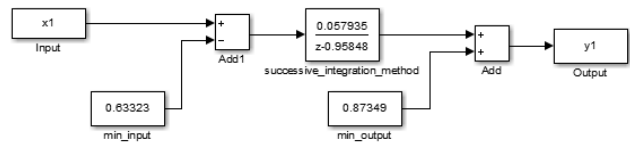
**Fig. 5** Block scheme of the area method (AM)



**Fig. 6** Time Delay Neural Network scheme (TDNN)

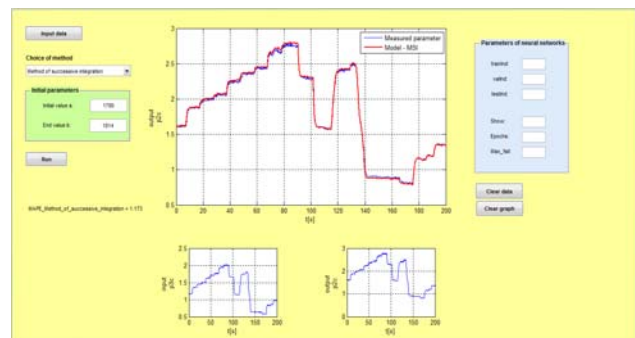
**4. TESTING THE IMPLEMENTED APPLICATION**

Testing of the application was done on data obtained from a small turbojet engine MPM-20 [13], [14] placed in the Laboratory of Intelligent Control Systems of Aircraft Engines. In deterministic methods, we used the  $P_{3C}$  as an input parameter (total pressure of gas on the input of the gas turbine) and as a requested output parameter  $P_{2C}$  (total pressure of air on the output from the radial compressor). After choosing the method of successive integration the calculation to obtain the transfer function takes place and a scheme of the resulting model is generated (Fig. 7).



**Fig. 7** Model scheme evaluated by a method of successive integration

Next, the resulting graph of the method of successive integration is drawn in the GUI. The result can be observed in Fig. 8. The dark blue color represents the progress of the measured parameter, in this case out chosen output, the  $P_{2C}$ . The red color represents the selected output of the model, thus the modeled values of the given parameter. On the left down side, the value of the mean absolute percentage error (MAPE) is shown. This value expresses the accuracy of the model, thus the deviation between the value of the measured parameter and the value evaluated by the model. Within this test, the error reached 1.173 %.



**Fig. 8** Model output shown in GUI of the application (MPI)

In neural networks, the  $Q_{pal}$  (weighted flow of the fuel) was designated as the input and  $P_{2C}$  as the requested output parameter. In the lower part of the application window, the progress graphs of the chosen parameter are drawn. Initial parameters and default values of parameters for neural networks are also automatically set (Fig. 9).

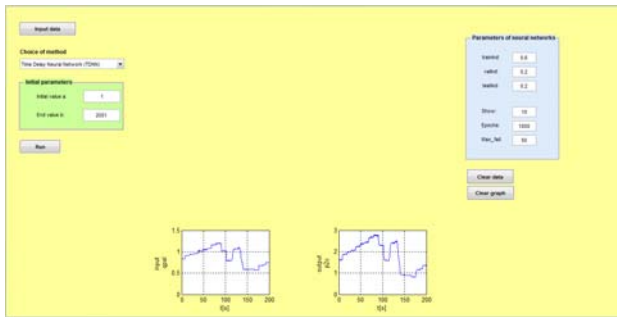


Fig. 9 Entered parameter values of the neural networks

We have chosen the Time Delay Neural Network and after the launch, the training of the neural network took place. As a result, an output experimental model was generated (Fig. 10).

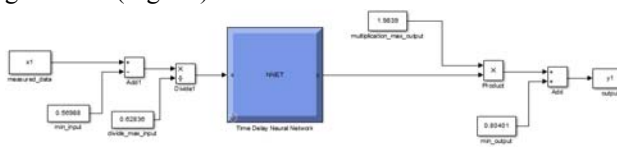


Fig. 10 Neural model scheme (TDNN)

Next, the resulting graph is drawn in the GUI. This graph can be observed in Fig. 11. The dark blue color line shows the progress of the measured parameter. It represents our chosen output  $P_{2C}$ . The black color represents selected output from the neural model (TDNN). The value of the mean absolute percentage error was evaluated as 2.3045 %.

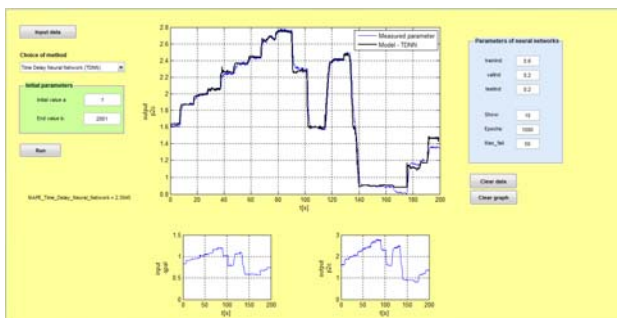


Fig. 11 Output of the model drawn in the GUI (TDNN)

In case we need to compare outputs of the models of each method based on the accuracy, the application allows us to draw the progress of the simulations without removing previous graphs and data. This means that all simulations can be shown in a single window, with color distinctions for each method and their errors. This view can be seen in Fig. 12, where the blue color shows the measured parameter (in this case the  $P_{2C}$ ), red is the output of the successive integration model, black is the output of TDNN and pink is the output of the DDNN.

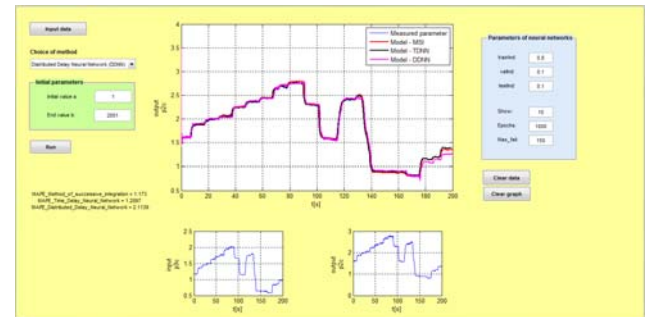


Fig. 12 Example of the GUI when comparing various methods

The best resulting models created by the proposed application was used in the diagnostic and backup system of the small turbojet engine [14], [15], [16].

## 5. CONCLUSIONS

This paper dealt with designing of an application for experimental identification of a system. MATLAB and Simulink programming environments were used. Five methods (Successive integration, Area method, Time Delay Neural Network, Distributed Delay Neural Network, NARX Neural Network) were described. From these methods we can pick any during the design of an experimental model. The application was tested on a dataset from a small turbojet engine MPM-20, where its functionality was approved.

The main benefits of the application are:

- work in a neat and user friendly graphical interface,
- possibility to choose from the five methods of experimental identification,
- automation of the process for creating experimental models, where the model's block scheme is automatically generated, its accuracy is evaluated and a graph from the outputs of the model is drawn,
- acceleration and simplification of the model design without the need to launch different scripts,
- evaluation and comparison of the experimental identification methods for the purpose of selection the most accurate method.

## ACKNOWLEDGMENTS

This work was supported by the Slovak Research and Development Agency under the contract No. APVV-16-0213 and by the Slovak Grant Agency of the Ministry of Education and Academy of Science of the Slovak Republic under grant No. 1/0493/16.

## REFERENCES

- [1] FIKAR, M. – MIKLEŠ, J.: Identifikácia systémov. STU Bratislava 1999. ISBN 80-227-1177-2. Available at: <<https://www.kirp.chtf.stuba.sk/~fikar/research/ident/ident.pdf>>.

- [2] LJUNG, L.: System Identification. New Jersey: PTR Prentice Hall, 1987, 255 pp., ISBN 0-13-881640-9.
- [3] NOSKIEVIČ, P.: Modelování a identifikace systémů. Ostrava: MONTANEX a.s., 1999. 276 pp., ISBN 80-7225-030-2.
- [4] FEDOROVÁ, D.: Experimentálna identifikácia zložitého systému v prostredí MATLAB/Simulink. Bachelor's thesis. Košice: FEI TUKE, 2015.
- [5] SUZUKI, K.: Artificial Neural Network: Architectures and Application. InTech, 2013, 264 pp., ISBN 978-953-51-0935-8.
- [6] TIMKO, J. – ŽILKOVÁ, J. – GIROVSKÝ, P.: Modelovanie a riadenie elektrických pohonov s využitím neurónových sietí. Košice: C-Press, 2009, 205 pp. ISBN 978-80-8086-124-7.
- [7] Neural Network's Role in Predictive Analytics. Available at: <<https://www.information-management.com/news/neural-networks-146-role-in-predictive-analytics>>.
- [8] KVASNIČKA, V. – BEŇUŠKOVÁ, E. – POSPÍCHAL J. – FARKAŠ, I. – TIŇO, P.: Úvod do teórie neurónových sietí. ISBN 80-88778-30-1. Available at: <[https://encyklopediapoznania.sk/data/eknihy/informatika/uvod\\_do\\_teorie\\_neurono\\_vych\\_sieti.pdf](https://encyklopediapoznania.sk/data/eknihy/informatika/uvod_do_teorie_neurono_vych_sieti.pdf)>.
- [9] KIM, Y. – KIM, Y.-S.: Optimizing neural network to develop loitering detection scheme for intelligent video surveillance systems. International Journal of Artificial Intelligence, Vol. 15, No. 2, 2017, pp. 30-39. ISSN 0974-0635.
- [10] BEALE, M. H. – HAGAN, M. T. – DEMUTH, H.B.: Neural Network Toolbox: User's Guide. The MathWorks, Inc., 2014. ISBN 0-9717321-0-8. Available at: <[http://www.mathworks.com/help/pdf\\_doc/nnet/nnet Ug.pdf](http://www.mathworks.com/help/pdf_doc/nnet/nnet Ug.pdf)>.
- [11] ZHANG, B. – LAM, J. – XU, S.: Stability Analysis of Distributed Delay Neural Networks Based on Relaxed Lyapunov–Krasovskii Functionals, in IEEE Transactions on Neural Networks and Learning Systems, Vol. 26, No. 7, 2015, pp. 1480-1492. ISSN 2162-2388.
- [12] GAO, Y. – ER, M. J.: NARMAX time series model prediction: feedforward and recurrent fuzzy neural network approaches. Fuzzy Sets and Systems, Vol. 150, No. 2, 2005, pp. 331-350. ISSN 0165-0114.
- [13] ANDOGA, R. – FŐZŐ, L. – MADARÁSZ, L. – KAROL, T.: A Digital Diagnostics System for Small Turbojet Engine. Acta Polytechnica Hungarica. Journal of Applied Sciences, Vol. 10, No. 4, 2013, 45-48, ISSN 1795-8860.
- [14] NYULÁSZI, L. – MADARÁSZ, L. – ANDOGA, R. – GAŠPAR, V.: Redundant Backup and Diagnostic System of MPM-20 Engine. CINTI 2013: 14th IEEE International Symposium on Computational Intelligence and Informatics: Proceedings: November 19-21, 2013, Budapest. - Piscataway: IEEE, 2013, 43-47, ISBN 978-1-4799-0195-1.
- [15] NYULÁSZI, L. – MADARÁSZ, L., “Experimental Identification of the Small Turbojet Engine MPM-20,” in CINTI 2014: 15th IEEE International Symposium on Computational Intelligence and Informatics: Proceedings: November 19-21, 2014, Budapest. - Piscataway: IEEE, 2014, pp. 497-501, ISBN 978-1-4799-5338-7.
- [16] MADARÁSZ, L., et al., “Komplexný výskum efektívnosti a inovácia technológie skúšok malého prúdového motora (4). In: AT&P Journal, Vol. 20, No. 12, 2013, pp. 49 – 52. ISSN 1335-2237.

Received November 27, 2017, accepted February 14, 2018

## BIOGRAPHIES

**Ladislav Nyulászi** graduated (MSc) with distinction at the Department of Cybernetics and Artificial Intelligence of the Faculty of Electrical Engineering and Informatics at Technical University in Košice. He is currently a PhD student at Department of Cybernetics and Artificial Intelligence. His scientific research is focusing on diagnostics and backup of complex dynamic systems, more precisely aircraft turbojet engines. In addition, he also investigates questions related with the datamining and efficiency analysis.

**Peter Butka** received his MSc degree in 2003 and PhD degree in 2010, both at the Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics at Technical University in Košice. His research interests include data/text mining, knowledge management, semantic technologies, and information retrieval.

**Vladimír Gašpar** graduated (MEng) with distinction at the department of Cybernetics and Artificial Intelligence. He is currently an assistant professor at Department of Cybernetics and Artificial Intelligence. His current research is aimed on mobile and embedded applications for stream processing and mining. The topic of his dissertation was to solve the problem of evaluating quality of experimental identification and operational quality of complex thermodynamic systems. He is currently involved in domestic research projects in fields of knowledge discovery, datamining, complex systems analysis and efficiency analysis.