The computer diagnostic program (DIAG 2) for identifying states of complex technical objects

Stanisław Duer¹, and Dariusz Bernatowicz^{2,*}

¹Technical University of Koszalin, Department of Mechanics, 15-17 Raclawicka St., 75-620 Koszalin, Poland ²Technical University of Koszalin, Department of Electronics and Informatics, 2 Śniadeckich St., 75-453 Koszalin, Poland

Abstract. This paper presents a method to control an operation process of a complex technical object, with the use of trivalent diagnostic information. Also, a general diagram of the complex technical object was presented, and its internal structure was described. A diagnostic analysis was conducted, as a result of which sets of the functional elements of the object and its diagnostic signals were determined. Also, the methodology of the diagnostic examination of the technical system was presented. The result was a functional and diagnostic model, which constituted the basis for initial diagnostic information, which is provided by the sets of information concerning the elements of the basic modules and their output signals. The theoretical results obtained in the present study were verified in practice on the example of a complex and reparable technical object. It belongs to the group of technical equipment for which a short time of shutdowns is required (an ineffective use of the object).

1 Introduction

Smart systems oriented to assist human technical and technological activities are widely discussed in literature. Such systems are used to supervise and organize technical and technological processes basing upon diagnostic and quality evaluation. In case of high complexity processes being determined by large number of variables and factor such systems become very helpful as they apply to states of technical objects such as equipment, devices, production resources but also to selection of mandatory conditions and parameters of a process. Important achievements in this matter were presented in the following works [1-6] Developed ideas and particular systems consisted of many specialized modules presented in details in these works practically relate to smart and complex systems used to supervise technological processes including operational processes. Those include, among others, studies conducted by B. Buchannan, E. Shortliffe, F. Hayer-Roth, D. Waterman, W. Pedrycz, W. Kacalak, and others [7-12], describing principles of smart systems' application to solve and aid organizational processes of different technical and technological problems. It is important to note that different methods of converting human-specialistknowledge into a form expressed by a computer programming are specified in those works. The works [13-17] provided by the author can be classified into those studies accordingly.

The paper by L. Będkowski, A. Rosiński and T. Dąbrowski [1] covers the theory of operation of technical devices. It includes a mathematical description of model of technical object considering its reliability and operation. The paper presents organization of the operation process with the use of object's models presented. The authors based their considerations upon wide experience in testing of technical devices as well as the conclusions driven during previous studies.

An early stage of realization of the renovation process is technical diagnosis of the object under operation.

It involves significant number of tasks necessary to be performed in order to determine and explicitly identify the technical states of the object in relation to a set of particular situations and events occurred or operational. Those actions are defined as identification – classification (diagnosis) of states [1,18-23].

From the very beginning, two-valued logic was applied within technical diagnosis, in which "1" is determined as Operational State and "0" as Nonoperational State. Such a diagnosis aims at identifying a nonoperational state (failure) and finding an element which is generating that state [19-21]. For the purpose of organization of the object's operational action, it is important to anticipate the states preceding nonoperational states; In such a case applying of twovalued logic has been proven insufficient [1,3, 21]. A technical diagnosis supplies the object's user with

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

Corresponding author: dariusz.bernatowicz@tu.koszalin.pl

diagnostic information in order to organize the operational process. A particular example of that can be a piece of information expressed in three-valued logic. If technical objects are diagnosed by means of three-valued logic, it gives an opportunity to identify the following states of the object: $operational - \{2\}$, nonoperational - $\{0\}$, as well as incompletely operational – $\{1\}$ [19-21]. Due to ability of recognizing an incompletely operational state of internal elements of the object the diagnostic information obtained from the object with this approach becomes, , a framework whether to make operational decisions or build operational strategies that organize new course of technical objects' operation called a state-based operation [3,4,9,13]. Where those two disciplines, that is a diagnostic theory and a theory of the technical object's operation, work together within organizing of the operational process of technical objects and optimizing the operation system.

Expected increase of available technical information is achieved by applying three-valued logic that introduces intermediate and incompletely operational states [3,7,16,18,22]. The complexity and importance of operations for modern technical object impose whole new demands to reliability and quality diagnosis. Diagnostic information provided to the user must be precise and reliable and most importantly quickly accessible. One of the methods used to counteract such problems is to apply multi-valued logic with a greater number of states [3,21]. Another reason for applying such approach is relatively low increase in calculative cost of expanding the number of distinguished states. Using multi-valued logic requires, however, many technical and economic analyses, as well as standardization of principles and rules applicable for different types of technical objects.

Research on application of four-valued logic within the diagnostic process of complex technical objects conducted by the authors resulted in creating computer diagnostic program (DIAG 2) that is a development of the intelligent system (DIAG 2). A complete multivalued logic diagnostic process was implemented within discussed computer program. The following tasks were mandatory: creating of a diagnostic-functional diagram, defining of a set of diagnostic information, as well as classifying and concluding [3,20,21].

2 Description of the diagnostic program (DIAG 2)

The main purpose of creation of new computer diagnostic program was a necessity to use multi-states logic and to open a possibility to develop it for additional tasks in the near future. The increase of functionality or integration of additional tasks in a form of independent modules should lead to a smart system that can aid the renovation process of operational characteristics of technical states [4]. Taking the above into account, it was mandatory to apply the most accurate parameterization and highest tightness of condition as possible for tasks carried out when developing the program. The computer diagnostic program (DIAG 2) is a "standalone" type program with Graphical User Interface (GUI), which was created using a popular object programming Java language. Using Java and Java Virtual Machine (JVM) necessary to run an application allowed to reach independence from any operating system. It relates to scientific communities, where a number of less known and popular operating systems are used. Despite the fact that JVM is the indirect runtime environment, staged operational nature of the program inconsiderably affects the performance in relation to applications created using native languages.

A similar approach was also applied when selecting a data format for both the project's structure specification and the output data of the classification process. By applying commonly used Extensible Markup Language standard (XML), it was possible to gain high flexibility and comfort while editing or saving data. Such data having a form of specified structures can be recorded both in files (documents) and assigned to XML data bases using XQuery [10]. In addition, XML standard allows to easily modify created data structures if further development is necessary.

Programme (DIAG 2) realizes a complete diagnostic process and consists of separate stages that are available in consecutive panels. Respective stages include the following tasks:

- specify a model of the technical object and based of which to create structure and diagnostic and functional diagrams,
- define diagnostic information of the examined object in a diagnostic subsystem by testing a state, analyzing a model and observing real operation processes,
- create an internal structure of the object in a form of diagnostic signals table to enable further calculations,
- classify states of the examined object using multivalued logic (two-, three-, or four-valued) within the specified ranges of states and parameters of radial basis function networks,
- make conclusions from the diagnostic process by analyzing the gained results in a form of state maps, graphs and protocols and a comparison of classification results for different cases of multivalued logic of states.

The diagnostic program being described is also characterized by a compact structure enabling to easily access the individual stages at any time of using it, clear GUI and an intuitive user interface. Most of the options relating to individual stages are accessible dually in the main menu or toolbars but also in the context menu with commands characteristic for a respective panel. All schemes and graphs can be separately printed out or saved on data storage devices in a form of graphic files.

3 Structure of the project and the diagnostic and functional diagram

Application of a hierarchic structure of the model in the system enabled to analyze complex technical objects by decomposing its components within one project. In the structure of the project depicted in Fig. 1 it can be considered more than one object, where another object can be a detailed analysis of any element of the previous one. This leads to greater depth of the object examined and tightness of its components. An internal structure of each object is divided into four levels of operational structure [4]: level one: object {O}, level two: functional units {E_i} in an object, level three: components {e_j} in each unit and level four: basic elements {e_{i,j}}.



Fig.1. Structure of the project applied in (DIAG 2).

Functional units are designated in the system as units and primary elements as elements, whereas components of the 3rd level act as intermediate "elements" allowing bidirectional transformation of a hierarchic form of the object into matrix of internal structure shown in Table 1.

Table1. An example of the object's internal structure.

Object's units	Fundamental elements $\{e_{i,j}\}$ in the object						
E ₁	e _{1,1}		$e_{1,i}$		$e_{1,I}$		
÷	:		÷		:		
Ei	e _{i,1}		e _{i,j}		Ø		
÷	:		÷		:		
EI	$e_{I,1}$		$e_{I,j}$		$e_{I,J}$		

where:

 $(e_{i,j})-j^{th}$ element within i^{th} assemble (module) of the object, E_i-i^{th} assemble of the object, \varnothing is the element which complements the dimension of the table.

The above method to divide an operational structure of the object ensures a convenient form of describing (addressing) the elements within the structure. In addition, it also enables to change a structure of the examined object at any time, including a change of number of units or primary elements in a specific unit.

The first stage for a diagnostic elaboration is to create not only a structure of the technical object and also its functional-diagnostic diagram which can be used to determine a set of diagnostic signals $\{X_{i,j}\}$. The object's functional units being a part of the functional-diagnostic diagram shown in Fig. 2 are addressed as (E_i) , where "i" stands for a number of a unit in the object. Individual elements of a unit are addressed as $(e_{i,j})$, where "j" stands for a number of an element in i-unit.

However, it is necessary to create a new project and objects to be examined prior to realizing respective stages of the diagnostic process. If any project is created or currently open, it will be closed and saved as a proper XML document. Opening a new project sets up all default parameters. If it is necessary to implement any change in parameters, it can be achieved using the option Preferences. All newly created projects are saved in the default directory "Data" in the main path of the program. When creating a project, as well as other elements of the structure such as objects, units and elements, it is required to fill in respective forms in the dialog box. Each form consists of specific information on individual elements of the structure, that is to say, full or abbreviated name, description, code, author, date, etc. If an existing project is opened, the whole structure, object's diagrams, input data and configuration variables are properly set up. (Fig. 2 and Fig. 3) [3,20-21].

The window shown in Fig. 2 divides the application into the basic parts: the left-hand part containing a hierarchic structure of the project, including its elements, and the right-hand and central one realizing consecutive stages of the diagnostic process. The left panel is the main panel being available at any time and fully integrated with particular stages of the central panel. It means that any changes in structure of the project are automatically updated in the central panel.



Fig. 2. Structure of the Project and a functional-diagnostic diagram of an electronic device to control an automotive engine.

A course of the diagnostic process with its preferences is considered independently for each object. It should be noted that each object, showing tight and unique features, enables to diversify a level of diagnostic depth of individual elements. If another object is chosen or a new one is created, previous results are saved and a new diagnostic process is being edited.

Stages of the diagnostic process are placed in separate panels with descriptions using tabs in the bottom of the central panel (Fig. 3).

The following tabs can be distinguished:

- Block Diagram enables to create functionaldiagnostic diagrams and specify diagnostic signals on the level of a functional unit for each object (Fig. 2),
- Input Data enables to insert measurement data,
- nominal and diagnostic signals using a form and a data file, or a file with data generated in a random manner with preassigned parameters,
- Signals defines a diagnostic signals table in accordance with an internal structure of the object that is used as an input structure in a classification

process. This panel also includes additional signal calculations,

- Classification – realizes the classification process of stages of the examined object depending on a type of logic used and a range of relevant, irrelevant and inadmissible variations. It enables to generate results of the classification process in a form of charts, graphs or reports.



Fig. 3. Inserting input data of diagnostic signals using a form.

4 Determination of the diagnostic information on the example of a car engine

The method for the diagnosis of technical object presented was verified on the example of a reparable technical object. The analogue controller unit for combustion automotive engine with its peripheries was used for that purpose (Fig. 4). Research set-up was developed on the basis of a spark ignition engine with multi-point injection MPF1. The object was subjected to a diagnostic analysis in result of which functional-diagnostic diagram was developed. In the example, an object was used whose internal structure (Fig. 4) is composed of seven modules ($E_1, E_2,..., E_7$) (Table 2), and each one of them, up to five elements were distinguished [3].



Fig. 4. Diagram of an electronic controller for an automotive engine, where: E_1 – ignition module, E_2 – fuelling module, E_3 – air-feeding module, E_4 – starting circuit, E_5 – power supply circuit, E_6 – engine block, E_7 – electronic control unit.

Assembly of the object	Structure of the object $\{e_{i,j}\}$						
Ei	e ₁	e ₂	e ₃	e ₄	e ₅		
E ₁	e _{1,1}	e _{1,2}	e _{1,3}	e _{1,4}	e _{1,5}		
E ₂	e _{2,1}	e _{2,2}	Ø	Ø	Ø		
E ₃	e _{3,1}	e _{3,2}	e _{3,3}	Ø	Ø		
E4	e _{4,1}	e _{4,2}	Ø	Ø	Ø		
E ₅	e _{5,1}	e _{5,2}	Ø	Ø	Ø		
E ₆	e _{6,1}	e _{6,2}	Ø	Ø	Ø		
E ₇	e _{7.1}	Ø	Ø	Ø	Ø		

where:

 E_1 – ignition module: $e_{1,1}$ – automotive alternator, $e_{1,2}$ – voltage regulator, $e_{1,3}$ – battery, $e_{1,4}$ – coil ignition, $e_{1,5}$ – sparking plug; E_2 – fuelling module: $e_{2,1}$ – fuel tank ventilation valve, $e_{2,2}$ – fuel injector; E_3 – air-feeding module: $e_{3,1}$ – air flow meter, $e_{3,2}$ – throttle position sensor, $e_{3,3}$ – idle run position controller; E_4 – starting circuit: $e_{4,1}$ – combustion knocking sensor $e_{4,2}$ – coolant temperature sensor; E_5 – power supply circuit: $e_{5,1}$ – oxygen sensor (1), $e_{5,2}$ – catalyser, $e_{5,3}$ oxygen sensor (2); E_6 – engine block: $e_{6,1}$ – crank shaft position sensor, $e_{6,2}$ – EGR valve: E_7 – $e_{7,1}$ electronic control unit, \emptyset is the element which complements the dimension of the table.

5 Classification of technical objects' states in intelligent diagnostic systems with using multi-valued logic

An external reflection of all states of the object is precisely measured values of physical variables that characterize the object. Such a variables called attributes are connected with technical parameters of the examined object. Considering above output signals of the object are highly dependent upon states of input signals and states of its primary elements (Fig. 5).



Fig. 5. A panel presenting a set of diagnostic signals and a table with additional computations.

A set of functional elements $\{e_{i,j}\}$ determined during diagnostic elaboration is a basis to put together diagnostic signals in a form of a table. An example table with diagnostic signals is shown in Fig. 5.

Before we pass to the panel "Signals", the tab "Input data" should be chosen in order to determine a set of signals as it enables to insert values of diagnostic and nominal signals. A diagnostic state of the object is determined on a basis of examination of a set of output signals (diagnostic signals) and it significantly depends on the results of comparison between an image of a diagnostic signal and the image of its model signal (nominal signal). A classification stage within the diagnosis process determines therefore a specified state for each primary element of the examined technical object. It is then verified whether values of output signals of particular elements are contained in a range of permissible variations. The classification of states whether for functional units or the object itself is realized basing upon the classification rules and it depends on a type of logic (a number of states) (Fig. 6).



Fig. 6. The classification panel of the technical object within the diagnostic process.

The process of states classification is realized after choosing the panel "Classification". This panel contains configuration options and results tables of states (states maps). Within the configuration the following items can be defined: a type of logic of states and their ranges, an option of the graphical presentation in the states table and RBF parameters. In case of multi-valued logic acceptable states of the object are determined within the Polish standards for technical reliability and the quality of operation [3, 20].

In the described system the maximum number of states is limited to four, which therefore it enables to choose two-, three-, or four-valued logic. Symbols 0,1,2, and 3 are assigned to states as a result of classification. In case of four-valued logic, those symbols specify the following states [18]:

- Operational State marked as "3", which determines the state in which it the object performs in accordance with its destination (green) (Fig. 6),
- Incompletely Operational State marked as "2", which determines the state of the object characterized by its incomplete operation (blue) (Fig. 6),
- Critically Operational State marked as "1", which determines a state of the object characterized by extremely limited operation (yellow) (Fig. 6),
- -Nonoperational State marked as "0", which determines the state in which the object is no longer functional and incapable to operate at all (red) (Fig. 6).

6 Conclusions

The paper describes the computer diagnostic program (DIAG 2). Project's structure and its individual elements such as model of the technical object, and creation of diagnostic-functional scheme are described. The paper also presents the method for determining diagnostic signals of the examined object and creating an internal structure of the object in a form of diagnostic signals tables. A significant part of the paper is a description of the classification process of the examined object's states in multi-valued logic realized with specified ranges of states and RBF parameters. An innovative part of the work is that diagnostic conclusions making (analysis) realized in multi-valued logic and ways of presenting of the results gained.

References

- 1. L. Bedkowski, T. Dabrowski, *Basic of the maintenance theory p. 2.* WAT, Warsaw, p. 187, (2006)
- 2. S. Duer, Def. Sci. J. 60, 5, (2010)
- 3. S. Duer, Intelligent system of supporting the process renewal of operating characteristics in complex technical objects, Technical University of Koszalin, Koszalin, p. 242 (2012)
- 4. Z. Palkova, Warsaw U. Lif. Sci. 55, (2010)
- 5. A. Rosiński, Reliability, R. Saf. Theory App. 3, (2010)
- 6. A. Rosiński, Adv. in Saf. Reli. R. Man. (2012)
- B. Buchannan, E. Shortliffe, *Rule Based expert* systems, Addison – Wesley Publishing Company, p. 387, (1985)
- 8. F. Hayer-Roth, D. Waterman, D. Lenat, *Building* expert systems, Addison – Wesley Publishing Company, p. 192, (1983)
- 9. W. Kacalak, M. Majewski, J.M. Zurada, Lecture Not. in Art. Int. 6114, (2010)
- W. Kacalak, M. Majewski, Lecture Notes in Computer Science 7666, Part IV, Springer 2012, 115-122, (2012)
- W. Pedrycz, *Fuzzy Control and fuzzy systems*. John Walley End Sons, Inc. p. 432, (1993).
- D. Waterman, A guide to export systems, Addison Wesley Publishing, (1986)
- 13. S. Kobayashi, K. Nakamura, *IEEE Expert*, Oct. (2011)
- P. Linz, R. Wang, Exploring Numerical Methods: An Introduction to Scientific Computing using Matlab, University of California, Davis, p. 172, (2003)
- 15. P. Linz, An Introduction to Formal Languages and Automata, University of California, Davis, (2002)
- 16. Z. Palkova, T. Rodny, Slovak U. of Agri. (2010)
- 17. Tresp, R. Molitar, European Con. on Art. Int. (1998)

- 18. M. Bialko, *The Basic properties of neural networks* and hybrid expert system, Tl University of Koszalin, Koszalin, p. 341, (2000)
- 19. S. Duer, R. Duer, Neural Comp. App. 19, 5, (2010)
- 20. S. Duer, Neural Comp. App. 21, 1, (2012)
- 21. S. Duer, R. Duer, S. Mazuru, Romanian Ass. Nonc. Tech (2016)
- 22. M. Majewski, W. Kacalak, Polish J. Envir. Stu. 18, 3B, (2009)
- 23. L. Pokorádi, S. Duer, Int. Sci. Con. Adv. Mech. Eng. (2016)