PAN ARCHIVES OF ACOUSTICS
Vol. **41**, No. 3, pp. 449–460 (2016)

DOI: 10.1515/aoa-2016-0046

Decision Support System for Identifying Technical Condition of Combustion Engine

Adam DEPTUŁA(1), Piotr OSIŃSKI(2),
Urszula RADZIWANOWSKA(2)

(1) *Faculty of Production Engineering and Logistics*
Opole University of Technology
Ozimska 75, 45-233 Opole, Poland

(2) *Faculty of Mechanical Engineering*
Wrocław University of Technology
Łukasiewicza 7/9, 50-371 Wrocław, Poland;
e-mail: urszula.radziwanowska@pwr.edu.pl

(*received November 27, 2015; accepted March 3, 2016*)

The paper is a continuation of the publication under the title "Acoustic diagnostics applications in the study of technical condition of combustion engine" and concerns the detailed description of decision support system for identifying technical condition (type of failure) of specified combustion engine. The input data were measured sound pressure levels of specific faults in comparison to the noise generated by undamaged motor. In the article, the whole procedure of decision method based on game graphs is described, as well as the interface of the program for direct usage.

Keywords: game graphs; acoustic diagnostics; combustion engine.

1. Introduction

The non-invasive diagnostics of internal-combustion engines are very interesting and desirable way to determine the occurring problems, because of the fact that they do not involve interference in the structure or disassembly of construction to identify a specific failure (Luft, 2010). Several methods of non-invasive engine testing were already proposed including the measurements of pressure and vibrations (Zhen *et al.*, 2013; Barelli *et al.*, 2008; Carlucci *et al.*, 2006; Ranachowski, Bejger, 2005). It is important to test engine operating parameters during the manufacturing process as well as in its subsequent operation (Teodorczyk, 2010). This allows to determine the optimal parameters of engine operation. The development of acoustic diagnostics is a promising method for quick and simple identification of engine damage. This method is nowadays widely used in a range of applications, especially in the field of machines testing (Kirpluk, 2012; Osiński, Kollek, 2013; Serdecki, 2012). The acoustic diagnostic method is based on the measurements of pressure wave spectrum, in particular the noise generated by engine during its exploitation. The detailed description of acoustic diagnostic method and results of measurements were described in previous article titled "Acoustic diagnostics applications in the study of technical condition of combustion engine". The aim of present publication is to present an algorithm of decision support system for identifying engine failures. In the literature, some other signal analysis algorithms related to fault diagnostics may be found (Wu, Liu, 2009; Wu, Chen, 2006; Adaileh, 2013; Elamin *et al.*, 2010; Ettefagh *et al.*, 2008; Głowacz, Kozik, 2013; Głowacz, Głowacz, 2010; Głowacz, 2014; 2015; Wu, Kuo, 2009a; 2009b; Sobański, Orłowska-Kowalska, 2015). Proposed in this paper decision method based on game graphs is a powerful tool for failure detection using measured sound pressure spectrum of operating engine. Game graphs have been used as a generate-and-test (search) tool (Deptuła, 2015) and in the optimization of hydraulic properties of machine systems (Deptuła, Partyka, 2014; Deptuła, 2014).

2. Sound pressure measurements

The measurements on a Fiat diesel engine with common-rail system were performed in order to prepare the accurate diagnostic maps. Comparison of acoustic pressure level between not warmed-up and warmed-up engine was the first step. Then four different cases with induced engine damages were examined: disconnected boost pressure sensor, disconnected camshaft position sensor, disconnected injector No. 2 and disconnected fuel pressure sensor. Measurements consisted of comparing the sound pressure level of specific frequencies for motor operation with induced defects with the smooth operation of the engine. All the data were collected for two engine speeds 1,000 rpm and 2,000 rpm. Dedicated test rig and measuring equipment were used in order to achieve acoustic wave spectrum for different cases of induced engine faults, including modular sound pressure level meter with the record time history and analysis of the frequency with preamplifier and microphone. More detailed description of measurements procedure may be found in the article "Acoustic diagnostics applications in the study of technical condition of combustion engine".

3. Method of decision trees

In order to create the method of decision trees it is possible to use heuristic search methods (Divina, Marchiori, 2005). Actions undertaken while the problem is being solved can be classified as searching of objects of the specified characteristics. For the identification of the type of engine damage, rules of procedure by the generate-and-test method focused on testing of a given property of an acoustic signal have been used.

An oriented graph can be defined as an ordered pair of sets. The graph vertices are included in the first one, whereas graph edges, that is an ordered pair of vertices, are included in the second one (Fig. 1).

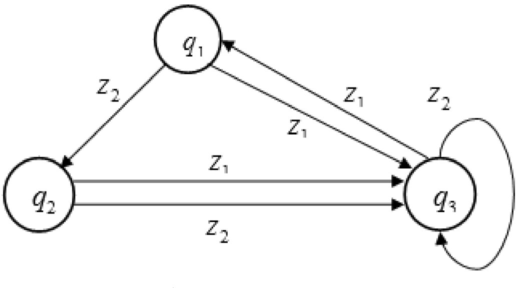


Fig. 1. An oriented game graph.

The oriented game graph shown in Fig. 1 is composed of a set of vertices Q:

$$Q = \{q_1, q_2, q_3\} \tag{1}$$

and of a set of edges Z, that is an ordered pair of vertices:

$$Z = \{z_1, z_2\}. \tag{2}$$

As a result of a graph distribution from the chosen vertex, a tree structure with cycles is obtained in the first step and then, a general game tree structure is obtained. Each of them has an appropriate analytical formulation G_i^+ and G_i^{++}.

A game tree structure is a part of the systematic searching method. The algorithm of the game graph distribution on the game structure is presented among others in papers (Deptuła, Partyka, 2011). A start vertex q_i is chosen in the first step. Acting in accordance with the algorithm and assuming that the start vertex is q_1, it is possible to transform the oriented dependence graph presented in Fig. 1 into an analytical formulation $G_{q_1}^+$, and then we obtain the following formulation as a result of the operation (3)

$$\begin{aligned} G_1^+ = ({}^0q_1({}^1z_1q_3({}^2z_1q_1, z_2q_3)^2,\\ z_2q_2({}^2z_1q_3, z_2q_3)^2)^1)^0. \end{aligned} \tag{3}$$

It is possible to return to an earlier vertex and even to a start vertex from an appropriate end vertex, so we obtain a game tree structure presented in Fig. 2.

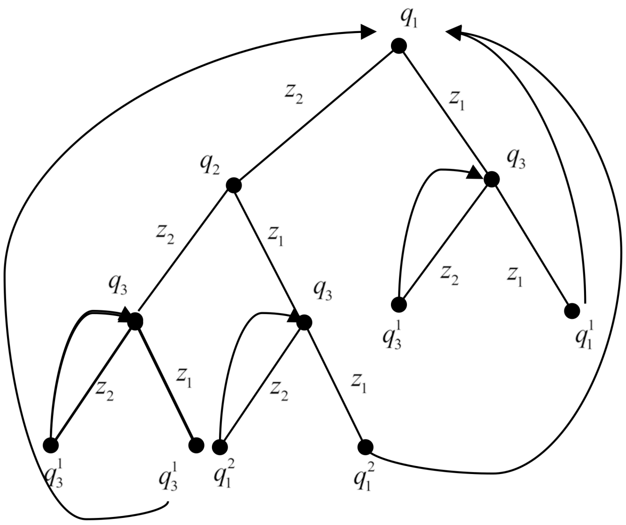


Fig. 2. A game tree structure with cycles with the start vertex q_1.

When we change cycles in the structure shown in Fig. 2 into decisions and/or expressions we obtain a game tree structure presented in Fig. 3.

Four terms are defined in order to create a decision support system: elements of the system, relations, properties of elements and the objective function.

It can be assumed that elements of the system form the set Q, relations – the set Ω, properties of elements – the set Γ, the objective function – F.

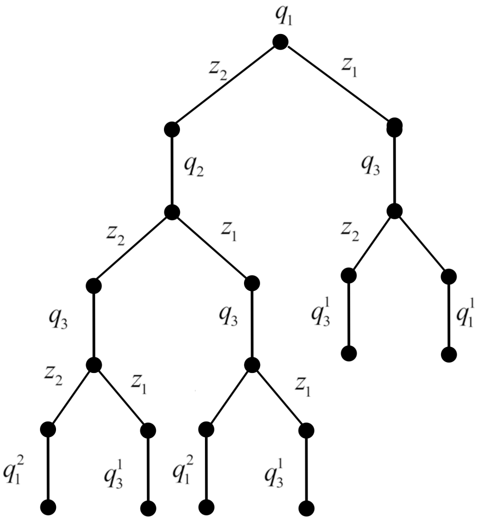


Fig. 3. A game tree structure with the start vertex q_1.

It can be assumed that states of the engine are elements of the set:

$$Q = \{X_0, X_1, X_2, \ldots, X_i, \ldots, X_m\}, \tag{4}$$

where X_i – i-th engine damage, if Q is a finite set.

For a given object $X_i \in Q$ it is possible to differentiate a certain level of sound pressure $\widetilde{A}_i$. Then, sound pressure values, as possible states of the analysed damage X_i will be a finite set in this domain

$$^+\widetilde{A}_i = \{a_{i,1}, a_{i,2}, ..., a_{i,j}, ..., a_{i,w}\}, \tag{5}$$

where elements $a_{i,j}$ for $i = \text{const}$ and $j = 1, 2, ...$, mean highlighted values in this domain (frequency) $\widetilde{A}_i$. Values of differences in the level of sound pressure ΔL for the frequency f [Hz]: {6.3 Hz, 8 Hz, 10 Hz, ..., 20 kHz, A, Z} will be included in the analysis.

Taking into consideration the applied description of a decision system, the state of the system $[Q]$ in a given domain is a vector in the following form:

$$a = [a_{1,j1}, a_{2,j2}, ..., a_{i,ji}, ..., a_{m,jm}], \tag{6}$$

where the coordinate $a_{1,j1}$ represents a damage X_1 in the domain $\widetilde{A}_1$, the coordinate $a_{2,j2}$ represents a damage X_2 in the domain $\widetilde{A}_2$, the coordinate $a_{1,j2}$ represents a damage X_2 in the domain $\widetilde{A}_1$.

It is necessary to define the relation of the measured values of sound pressure levels values of given damages X_i in order to differentiate damages. The set Π_s composed of the ordered pairs (X_i, X_j) can be called the relation on the set Q. The set Π_s is a subset of the Cartesian product $Q \times Q$ and has the following form:

$$\Pi_S = \{(X_{i1}, X_{j1}), (X_{i2}, X_{j2}), ..., (X_{is}, X_{js})\}, \tag{7}$$

where i_s, j_s are specific values of the sound pressure level of a given damage.

Relations of different kinds can occur among sound pressure values e.g. arithmetic (signs +, −), causal, logical, reciprocal and this is why the set of possible relation types in the decision system $[Q]$ among damages X_i is taken into consideration and it has the following form:

$$\Pi_S = \{\Pi_1, \Pi_2, ..., \Pi_S, ..., \Pi_r\},$$

where $\Pi_s \in \Pi$ is a set from the expression (7).

An example of a graph of relations Π_s for damages $X_i = X_5$ is presented below.

For the analysis of a decision making system in the evaluation of the current state of relations (and then the kind of damage), each pair $(X_i, X_j) \in \Pi_S^i$ occurring in the relation $X_i \Pi_S X_j$ has to have a set of possible values which are taken by the relation Π_s between objects X_i and X_j. Such a set can be presented in the following form:

$$^+\Pi_s^{i,j} = \left\{\delta_s^{i,j}, \delta_{s,2}^{i,j}, ..., \delta_{s,k}^{i,j}, ..., \delta_{s,w}^{i,j}\right\}, \tag{8}$$

where $\delta_{s,k}^r$ is a possible highlighted value.

When searching a game graph, there will be a possibility of determining the kind of damage X_i in the scope of the relation of the Π_s type by means of defining highlighted values.

Such a state can be presented in the following form of a vector:

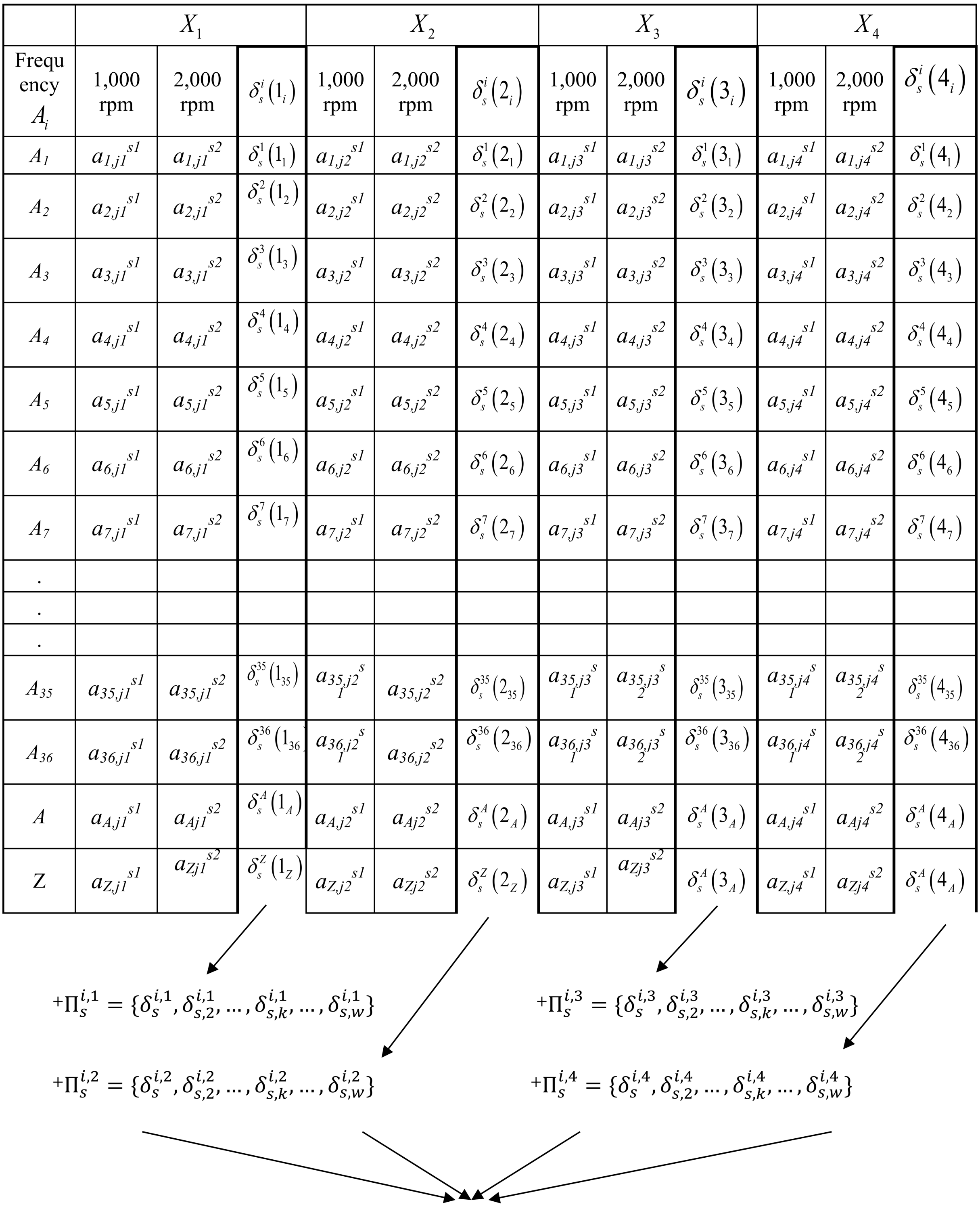
$$\Pi_s^i = \left[\delta_s^i(j_1), \delta_s^i(j_2), ..., \delta_s^i(j_n)\right]. \tag{9}$$

In case of our analysis, there is the following combination table (Fig. 4):

X_1 – disconnected boost pressure sensor,
X_2 – disconnected shaft position sensor,
X_3 – disconnected injector No. 2,
X_4 – disconnected fuel pressure sensor,
$\{A_1, A_2, A_3, ..., A_i, A, Z\}$ – frequency,
$a_{i,j}$ – sound pressure values of j-th damage in i-th domain,
$\delta_s^i(j_i)$ – a highlighted value.

It is possible to create a game graph for the above table as shown in Fig. 5.

The algorithm of the game graph distribution is connected with the choice of searching states (of the sound pressure values) of a given damage and "determining" the highlighted value set $\Pi_S = \{\Pi_1, \Pi_2, ..., \Pi_S, ..., \Pi_r\}$ by the graph. In this way, highlighted values $\delta_s^i(j_i)$ are determined for each damage X_1, X_2, X_3, and X_4. The search is made from the last checked vertex along the unsearched edge. The evaluation of nodes properties: $a_{1,j1}^{s1}$, $a_{1,j1}^{s2}$, $a_{3,j1}^{s1}$ ... etc., depends strictly on the properties of their parents (damages: X_1, X_2, X_3 and X_4). Greedy searching choses a locally optimum (the best at the moment) highlighted value in each step.



	X_1			X_2			X_3			X_4		
Frequency A_i	1,000 rpm	2,000 rpm	$\delta_s^i(1_i)$	1,000 rpm	2,000 rpm	$\delta_s^i(2_i)$	1,000 rpm	2,000 rpm	$\delta_s^i(3_i)$	1,000 rpm	2,000 rpm	$\delta_s^i(4_i)$
A_1	$a_{1,j1}^{s1}$	$a_{1,j1}^{s2}$	$\delta_s^1(1_1)$	$a_{1,j2}^{s1}$	$a_{1,j2}^{s2}$	$\delta_s^1(2_1)$	$a_{1,j3}^{s1}$	$a_{1,j3}^{s2}$	$\delta_s^1(3_1)$	$a_{1,j4}^{s1}$	$a_{1,j4}^{s2}$	$\delta_s^1(4_1)$
A_2	$a_{2,j1}^{s1}$	$a_{2,j1}^{s2}$	$\delta_s^2(1_2)$	$a_{2,j2}^{s1}$	$a_{2,j2}^{s2}$	$\delta_s^2(2_2)$	$a_{2,j3}^{s1}$	$a_{2,j3}^{s2}$	$\delta_s^2(3_2)$	$a_{2,j4}^{s1}$	$a_{2,j4}^{s2}$	$\delta_s^2(4_2)$
A_3	$a_{3,j1}^{s1}$	$a_{3,j1}^{s2}$	$\delta_s^3(1_3)$	$a_{3,j2}^{s1}$	$a_{3,j2}^{s2}$	$\delta_s^3(2_3)$	$a_{3,j3}^{s1}$	$a_{3,j3}^{s2}$	$\delta_s^3(3_3)$	$a_{3,j4}^{s1}$	$a_{3,j4}^{s2}$	$\delta_s^3(4_3)$
A_4	$a_{4,j1}^{s1}$	$a_{4,j1}^{s2}$	$\delta_s^4(1_4)$	$a_{4,j2}^{s1}$	$a_{4,j2}^{s2}$	$\delta_s^4(2_4)$	$a_{4,j3}^{s1}$	$a_{4,j3}^{s2}$	$\delta_s^4(3_4)$	$a_{4,j4}^{s1}$	$a_{4,j4}^{s2}$	$\delta_s^4(4_4)$
A_5	$a_{5,j1}^{s1}$	$a_{5,j1}^{s2}$	$\delta_s^5(1_5)$	$a_{5,j2}^{s1}$	$a_{5,j2}^{s2}$	$\delta_s^5(2_5)$	$a_{5,j3}^{s1}$	$a_{5,j3}^{s2}$	$\delta_s^5(3_5)$	$a_{5,j4}^{s1}$	$a_{5,j4}^{s2}$	$\delta_s^5(4_5)$
A_6	$a_{6,j1}^{s1}$	$a_{6,j1}^{s2}$	$\delta_s^6(1_6)$	$a_{6,j2}^{s1}$	$a_{6,j2}^{s2}$	$\delta_s^6(2_6)$	$a_{6,j3}^{s1}$	$a_{6,j3}^{s2}$	$\delta_s^6(3_6)$	$a_{6,j4}^{s1}$	$a_{6,j4}^{s2}$	$\delta_s^6(4_6)$
A_7	$a_{7,j1}^{s1}$	$a_{7,j1}^{s2}$	$\delta_s^7(1_7)$	$a_{7,j2}^{s1}$	$a_{7,j2}^{s2}$	$\delta_s^7(2_7)$	$a_{7,j3}^{s1}$	$a_{7,j3}^{s2}$	$\delta_s^7(3_7)$	$a_{7,j4}^{s1}$	$a_{7,j4}^{s2}$	$\delta_s^7(4_7)$
.												
.												
.												
A_{35}	$a_{35,j1}^{s1}$	$a_{35,j1}^{s2}$	$\delta_s^{35}(1_{35})$	$a_{35,j2}^{s}$ 1	$a_{35,j2}^{s2}$	$\delta_s^{35}(2_{35})$	$a_{35,j3}^{s}$ 1	$a_{35,j3}^{s}$ 2	$\delta_s^{35}(3_{35})$	$a_{35,j4}^{s}$ 1	$a_{35,j4}^{s}$ 2	$\delta_s^{35}(4_{35})$
A_{36}	$a_{36,j1}^{s1}$	$a_{36,j1}^{s2}$	$\delta_s^{36}(1_{36}$	$a_{36,j2}^{s}$ 1	$a_{36,j2}^{s2}$	$\delta_s^{36}(2_{36})$	$a_{36,j3}^{s}$ 1	$a_{36,j3}^{s}$ 2	$\delta_s^{36}(3_{36})$	$a_{36,j4}^{s}$ 1	$a_{36,j4}^{s}$ 2	$\delta_s^{36}(4_{36})$
A	$a_{A,j1}^{s1}$	a_{Aj1}^{s2}	$\delta_s^A(1_A)$	$a_{A,j2}^{s1}$	a_{Aj2}^{s2}	$\delta_s^A(2_A)$	$a_{A,j3}^{s1}$	a_{Aj3}^{s2}	$\delta_s^A(3_A)$	$a_{A,j4}^{s1}$	a_{Aj4}^{s2}	$\delta_s^A(4_A)$
Z	$a_{Z,j1}^{s1}$	a_{Zj1}^{s2}	$\delta_s^Z(1_Z)$	$a_{Z,j2}^{s1}$	a_{Zj2}^{s2}	$\delta_s^Z(2_Z)$	$a_{Z,j3}^{s1}$	a_{Zj3}^{s2}	$\delta_s^A(3_A)$	$a_{Z,j4}^{s1}$	a_{Zj4}^{s2}	$\delta_s^A(4_A)$

$$^{+}\Pi_s^{i,1} = \{\delta_s^{i,1}, \delta_{s,2}^{i,1}, \dots, \delta_{s,k}^{i,1}, \dots, \delta_{s,w}^{i,1}\}$$

$$^{+}\Pi_s^{i,3} = \{\delta_s^{i,3}, \delta_{s,2}^{i,3}, \dots, \delta_{s,k}^{i,3}, \dots, \delta_{s,w}^{i,3}\}$$

$$^{+}\Pi_s^{i,2} = \{\delta_s^{i,2}, \delta_{s,2}^{i,2}, \dots, \delta_{s,k}^{i,2}, \dots, \delta_{s,w}^{i,2}\}$$

$$^{+}\Pi_s^{i,4} = \{\delta_s^{i,4}, \delta_{s,2}^{i,4}, \dots, \delta_{s,k}^{i,4}, \dots, \delta_{s,w}^{i,4}\}$$

$$\Pi_S = \{\Pi_1, \Pi_2, \dots, \Pi_S, \dots, \Pi_r\}$$

Fig. 4. The morphological table.

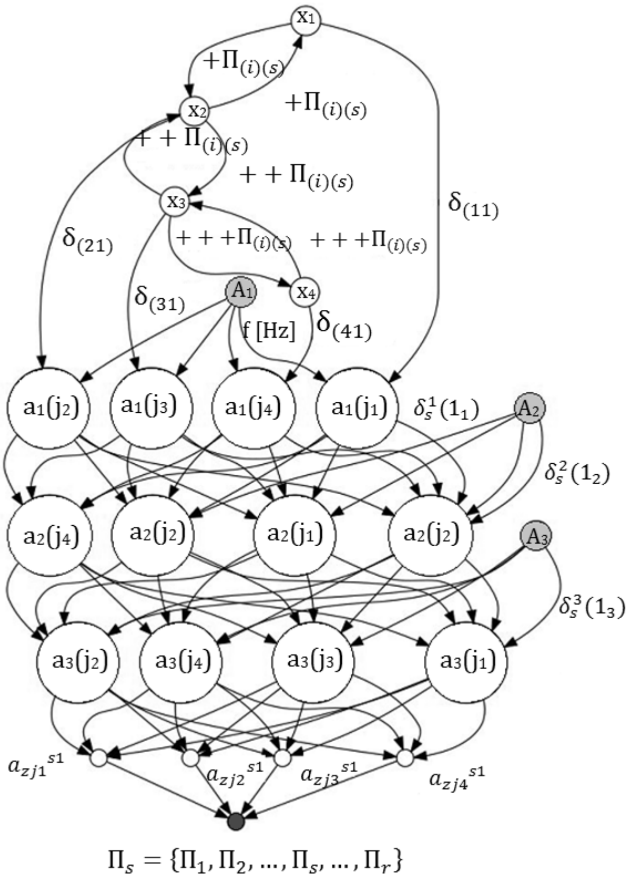


Fig. 5. A game graph for the morphological table.

4. The calculations of engine failure identification

In this section an analysis of damages in a combustion engine on the basis of the engine acoustic signals acquisition is described. In Figs. 6–9 the sound pressure levels comparison of different engine conditions are presented. Figures 6 and 7 compare warmed-up and not warmed-up engines operating with rotational speed of 1000 rpm and 2000 rpm respectively. Figures 8 and 9 show acoustic sound levels of specified failures: disconnected boost pressure sensor (failure I), disconnected camshaft position sensor (failure II), disconnected injector No. 2 (failure III), disconnected fuel pressure sensor (failure IV) in comparison to proper operation of engine.

Based on the acquired measurements results, the morphological table for this specific case was prepared as shown in Table 1.

For data included in morphological table (Table 1) it is necessary to create the game graph presented in Fig. 10. A game graph G as a "calculation unit" presented in Fig. 11 is analysed.

The sets of highlighted value presenting features of sound pressure values for a given damage have been obtained as a result of an analysis. The above summaries make it possible to create a measuring ladder, which will make it possible to determine the type of engine damage on the basis of sound pressure measurements on chosen frequencies.

For example the following formulation:

$$
{}^{+}\Pi_s^{i,1} = \left\{ \begin{matrix} \delta^1_{17,2}\left(a^{s1(-)}_{17,j1}, a^{s2(-)}_{17,j1}\right) \\ \delta^1_{20,2}\left(a^{s1(-)}_{20,j1}, a^{s2(+)}_{20,j1}\right) \\ \delta^1_{25,2}\left(a^{s1(+)}_{25,j1}, a^{s2(-)}_{25,j1}\right) \\ \delta^1_{30,2}\left(a^{s1(+)}_{30,j1}, a^{s2(-)}_{30,j1}\right) \end{matrix} \right\}
$$

means that the detection of the damage 1 will be based on the identification whether for the frequency 250 Hz for rotational speeds 1000 and 2000 $\delta^1_{17,2}(a^{s1(-)}_{17,j1}, a^{s2(-)}_{17,j1})$ there is a negative difference of sound pressures in relation to the pressures of an operating engine or whether there are alternate signs of sound pressure for the frequency 500 Hz and the rotational speeds 1000 and 2000 $\delta^1_{17,2}(\delta^1_{20,2}(a^{s1(-)}_{20,j1}, a^{s2(+)}_{20,j1}))$, therefore, in example (see Fig. 12).

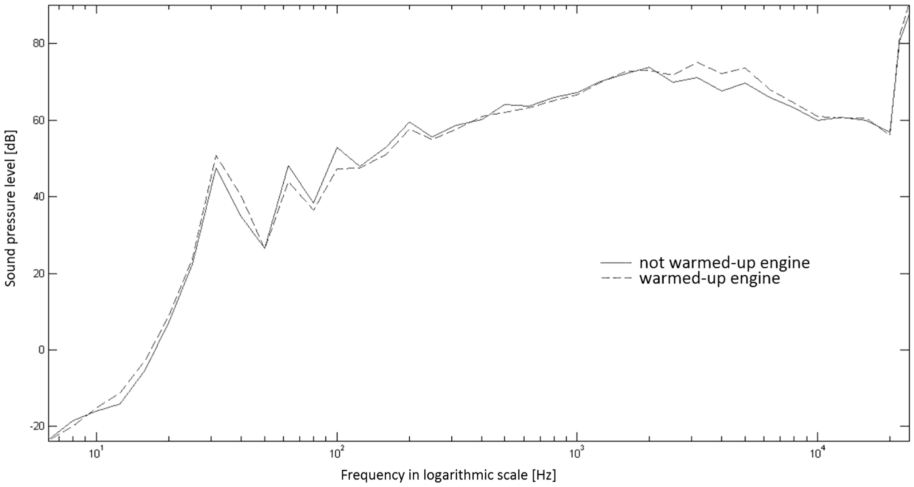


Fig. 6. Graph of sound pressure level for the speed of 1,000 rpm.

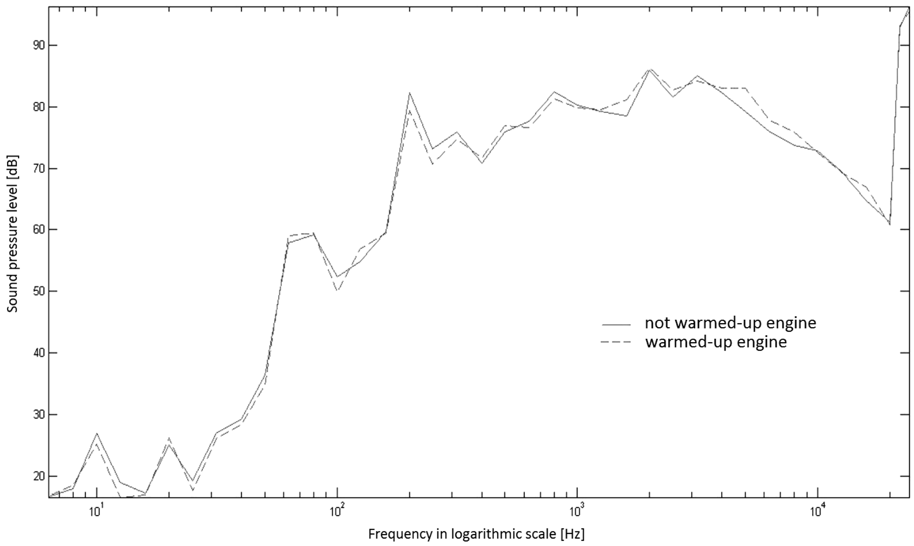


Fig. 7. Graph of sound pressure level for the speed of 2,000 rpm.

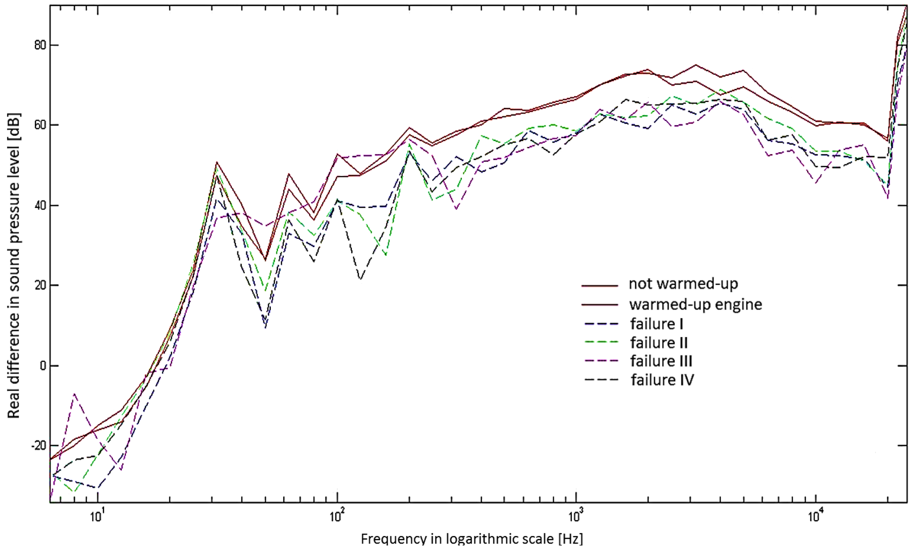


Fig. 8. Graph of sound pressure level for the speed of 1,000 rpm – all failures comparison.

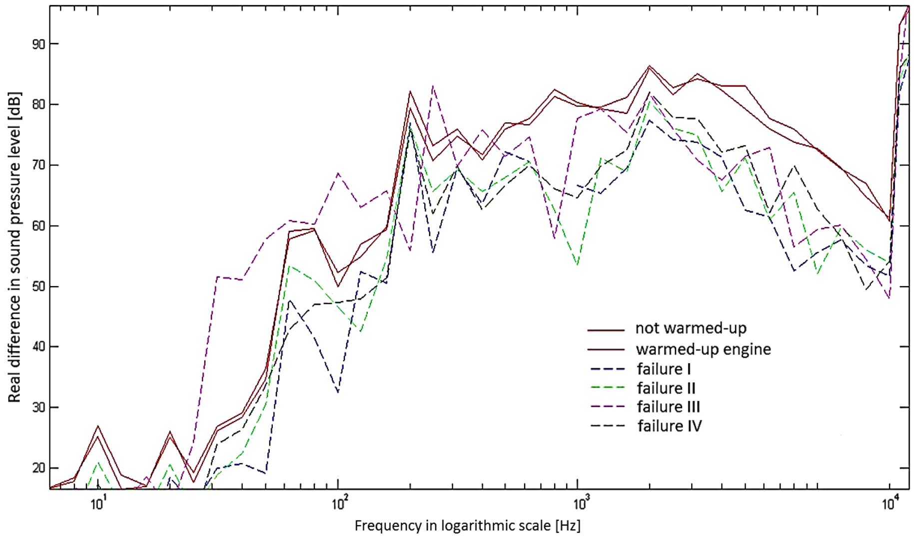


Fig. 9. Graph of sound pressure level for the speed of 2,000 rpm – all failures comparison.

Table 1. Morphological table of measurement results.

The difference in sound pressure

Damage number

		X_1		X_2		X_3		X_4	
Frequency		1000 rpm	2000 rpm	1000 rpm	2000 rpm	1000 rpm	2000 rpm	1000 rpm	2000 rpm
6.3 Hz	A_1	-2.53	1.09	-3.27	0.87	-0.42	1.31	-2.06	0.96
8 Hz	A_2	0.51	-0.58	0.28	-1.05	13.07	-4.99	-2.35	0.25
10 Hz	A_3	0.12	0.51	-0.95	1.4	1.56	0.42	0.72	0.67
12.5 Hz	A_4	0.28	0.44	2.33	1.85	0.14	1.75	1.58	1
16 Hz	A_5	0.72	-0.14	2.75	1.47	3.41	3.93	1.96	-0.43
20 Hz	A_6	0.78	-0.84	2.6	-1.39	0.47	0.2	1.7	-0.56
25 Hz	A_7	1.13	1.45	3.88	1.35	1.43	7.4	2.46	0.78
31.5 Hz	A_8	0.5	0.96	2.24	-0.89	0.17	25.4	1.66	2.07
40 Hz	A_9	-0.91	0.7	-1.08	1	2.07	22.65	0.11	2.16
50 Hz	A_{10}	0.09	0.12	-0.86	1.4	9.11	23	0.14	2.49
63 Hz	A_{11}	-0.35	0.32	-1.35	-1.36	-1.28	3.97	0.67	0.1
80 Hz	A_{12}	-1.05	-0.07	-2.21	0.56	5.77	3.38	-0.4	0.24
100 Hz	A_{13}	-1.2	0.08	-1.2	1.7	5.81	18.88	-1.37	1.91
125 Hz	A_{14}	0.64	1.31	-0.47	-0.16	6.06	7.03	0.01	0.52
160 Hz	A_{15}	-0.34	0.52	-0.02	-1.7	3.9	7.18	-0.1	-0.73
200 Hz	A_{16}	1.3	-3.76	-3.84	-3.1	2.44	-0.02	-2.07	-2.48
250 Hz	A_{17}	-0.62	-0.14	0.19	-1.63	1.93	12.65	0.29	-0.64
315 Hz	A_{18}	-1.41	-1.54	-0.19	-1.4	-0.06	-1.67	-0.73	-1.68
400 Hz	A_{19}	-0.25	-0.73	-2.57	-1.2	-0.45	5.48	-0.63	-0.55
500 Hz	A_{20}	-0.32	1.25	-1.06	-0.59	-0.46	-1.43	-1	-0.43
630 Hz	A_{21}	-1.9	0.97	-2.24	0.96	-0.63	2.11	-1.12	0.85
800 Hz	A_{22}	-0.56	0.001	-1.62	-0.06	-0.67	0.02	-0.26	0.13
1 kHz	A_{23}	-0.69	-0.22	-0.75	-0.01	0.5	2.07	-0.64	0.13
1.25 kHz	A_{24}	-0.91	0.16	-0.92	0.58	0.99	2.82	-0.57	0.43
1.6 kHz	A_{25}	0.25	-0.31	-0.36	-0.26	0.31	1.02	-1.12	-0.63
2 kHz	A_{26}	-0.18	-0.59	-0.38	-1.3	0.74	-1.85	-0.74	-2.02
2.5 kHz	A_{27}	-1.01	-0.66	-1.9	-1.08	-0.27	-1.04	-1.09	-1.74
3.15 kHz	A_{28}	-0.27	-0.4	-0.47	-0.54	0.16	-0.2	-0.51	-1.06
4 kHz	A_{29}	-1.1	-0.3	-2.86	0.08	-1.21	0.12	-1.38	-0.37
5 kHz	A_{30}	0.45	-0.04	-0.76	-0.29	0.33	0.29	-0.82	-0.48
6.3 kHz	A_{31}	-0.3	0.1	-1.16	0.09	-0.12	1.23	-0.29	-0.12
8 kHz	A_{32}	-0.56	-0.02	-1.49	-0.42	-0.39	-0.05	-1	-1.27
10 kHz	A_{33}	-0.69	-0.09	-0.86	0.04	-0.13	-0.22	-0.33	-0.48
12.5 kHz	A_{34}	-0.73	0.28	-0.98	0.45	-1.05	0.49	-0.35	-0.35
16 kHz	A_{35}	-0.56	0.19	-0.52	0.34	-1.4	0.24	-0.66	-0.08
20 kHz	A_{36}	0.29	0.51	0.33	0.81	-0.17	0.22	1.46	0.85
A	A_A	-0.31	-0.33	-0.95	-0.54	0.14	0.56	-0.78	-0.86
Z	A_Z	0.43	-0.76	2.06	-0.98	0.39	4.11	1.43	-0.91

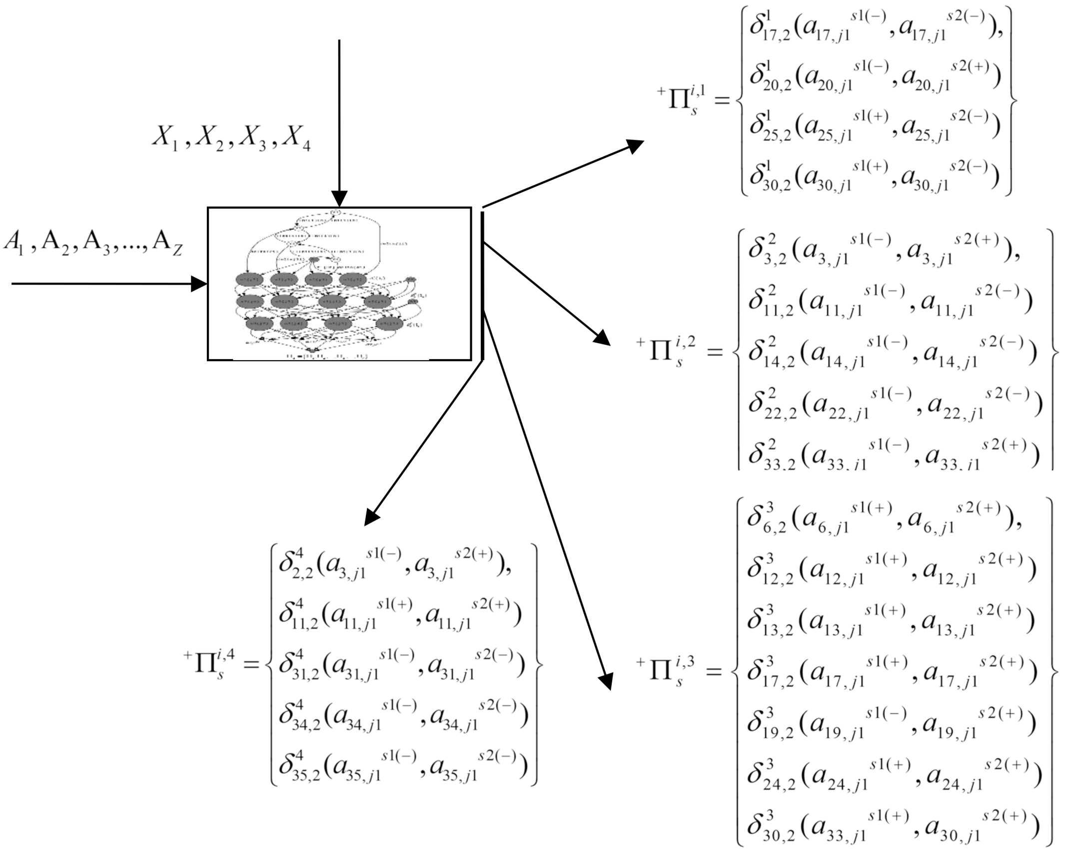


Fig. 10. Game graph for engine failure identification.

```
// parentsOf(x,G) is a list of the parents of node x in graph G

vector<Node> parentsOf(node x, graph G)
{ vector<Node> parents; // create empty list of nodes
  for (i=0; i< G.edges (). size(); i++)
    if(G.edges [i].destination ()==x)
      parents.push_back(G.edges [i]. source ());
  return(parents);
}
//  If G is a graph and C, P are lists of nodes, then //
nodesWNparentsInP(G, C, P, N) is a list of the nodes in C
// with exactly N parents in P.

vector<Node> nodesWNParentsInP(gtaph G, vector<Node> C, vector<Node> P, int N)
{ vector<Node> returnList;    //The return list begins as empty.
  for(i=0; i < C.size(); i++  // Then, for every node in C, ...
{ parents= parentsOf(C[i], G);
  count=0;
  // compare each of the node's parents to each node in P,
  // and count the number of matches.
  for(j=0 j < parents.size (); j++)
     for (k=0; k < P.size (); k++)
         if(parents [j]==P[k])
            count++;
  // If the final count is N, add the node to the return list.
  if(count==N) returnList.push_pack(C[i]);
}
return (returnList);
}
```

Fig. 11. A part of a graph search algorithm code and the scheme of calculation units.

$$
{}^{+}\Pi_s^{i,1} = \begin{Bmatrix} \delta^1_{17,2}\left(a^{s1(-)}_{17,j1}, a^{s2(-)}_{17,j1}\right) \\ \delta^1_{20,2}\left(a^{s1(-)}_{20,j1}, a^{s2(+)}_{20,j1}\right) \\ \delta^1_{25,2}\left(a^{s1(+)}_{25,j1}, a^{s2(-)}_{25,j1}\right) \\ \delta^1_{30,2}\left(a^{s1(+)}_{30,j1}, a^{s2(-)}_{30,j1}\right) \end{Bmatrix}
$$

Damage 1	1,000 rpm	2,000 rpm
Frequency [Hz]	Difference sign of the sound pressure	
250	-	-
500	-	+
1600	+	-
5000	+	-

$$
{}^{+}\Pi_s^{i,2} = \begin{Bmatrix} \delta^2_{3,2}\left(a^{s1(-)}_{3,j1}, a^{s2(+)}_{3,j1}\right) \\ \delta^2_{11,2}\left(a^{s1(-)}_{11,j1}, a^{s2(-)}_{11,j1}\right) \\ \delta^2_{14,2}\left(a^{s1(-)}_{14,j1}, a^{s2(-)}_{14,j1}\right) \\ \delta^2_{22,2}\left(a^{s1(-)}_{22,j1}, a^{s2(-)}_{22,j1}\right) \\ \delta^2_{33,2}\left(a^{s1(-)}_{33,j1}, a^{s2(+)}_{33,j1}\right) \end{Bmatrix}
$$

Damage 2	1,000 rpm	2,000 rpm
Frequency [Hz]	Difference sign of the sound pressure	
10	-	+
63	-	-
125	-	-
800	-	-
10000	-	+

$$
{}^{+}\Pi_s^{i,4} = \begin{Bmatrix} \delta^4_{2,2}\left(a^{s1(-)}_{3,j1}, a^{s2(+)}_{3,j1}\right) \\ \delta^4_{11,2}\left(a^{s1(+)}_{11,j1}, a^{s2(+)}_{11,j1}\right) \\ \delta^4_{31,2}\left(a^{s1(-)}_{31,j1}, a^{s2(-)}_{31,j1}\right) \\ \delta^4_{34,2}\left(a^{s1(-)}_{34,j1}, a^{s2(-)}_{34,j1}\right) \\ \delta^4_{35,2}\left(a^{s1(-)}_{35,j1}, a^{s2(-)}_{35,j1}\right) \end{Bmatrix}
$$

Damage 4	1,000 rpm	2,000 rpm
Frequency [Hz]	Difference sign of the sound pressure	
8	-	+
63	+	+
6300	-	-
12500	-	-
16000	-	-

Fig. 12. Detection of engine damage based on sign comparison.

If features such as signs do not characterise the belonging to a given damage, then it is possible to take into consideration values or deviations occurring in the measurement of the sound pressure level.

5. Computer – integrated decision support system

As the most advanced form of substantially integrated system can be integrated system, resulting from the integration of complex systems-aided management of complex systems supported decision-making.

All operations were made in the real time during normal exploitation of the motor housing. On the basis of the results received from the inference mechanism, resulting from the correlation between acoustic signals measured in a continuous manner and model signals placed in the data base, there would be identification of the influence of particular signals from microphone on the chosen parameters of the motor housing (Fig. 13).

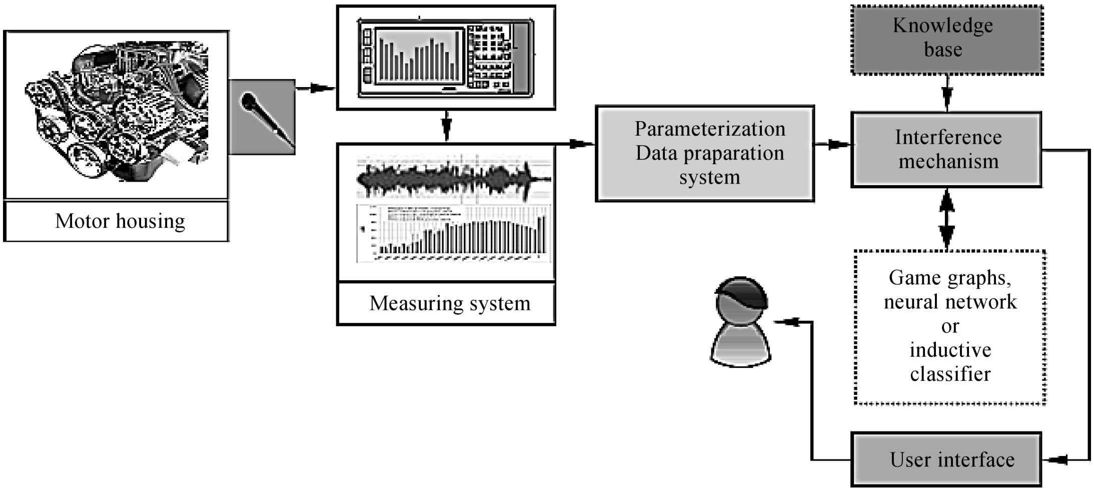


Fig. 13. Decision support system.

To create a decision support system used CAKE system (Computer Aided Engineering Knowledge) for the implementation of computer-aided application domain-system PC-Shell expert. The main tasks of the system is:

- automatic generation of a knowledge base in binary form, which is not subject to the process of translation, the appropriate structures inference module will be filled in automatically,
- defining the knowledge stored in binary system based on passwords and permissions,
- efficient and ergonomic knowledge base management through tools in the form of specialized editors blocks of knowledge bases.

The following algorithm is proposed (Fig. 14).

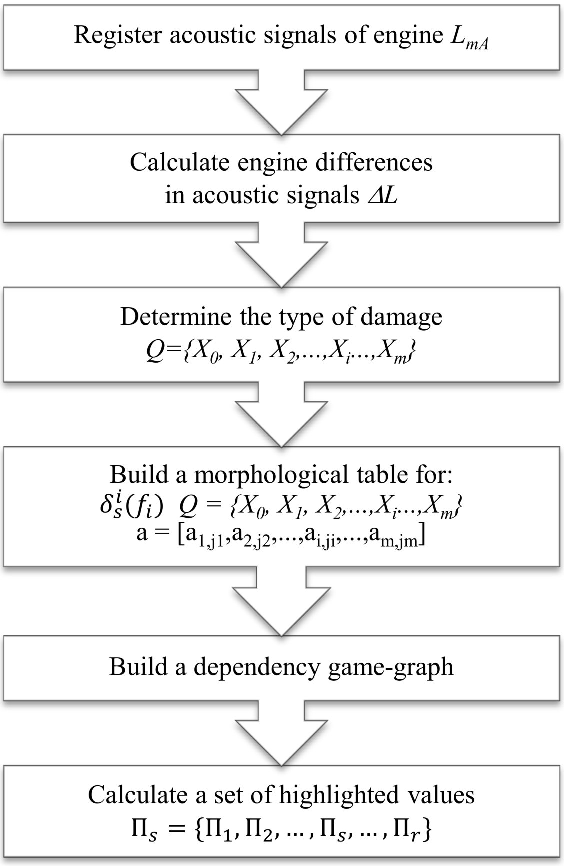
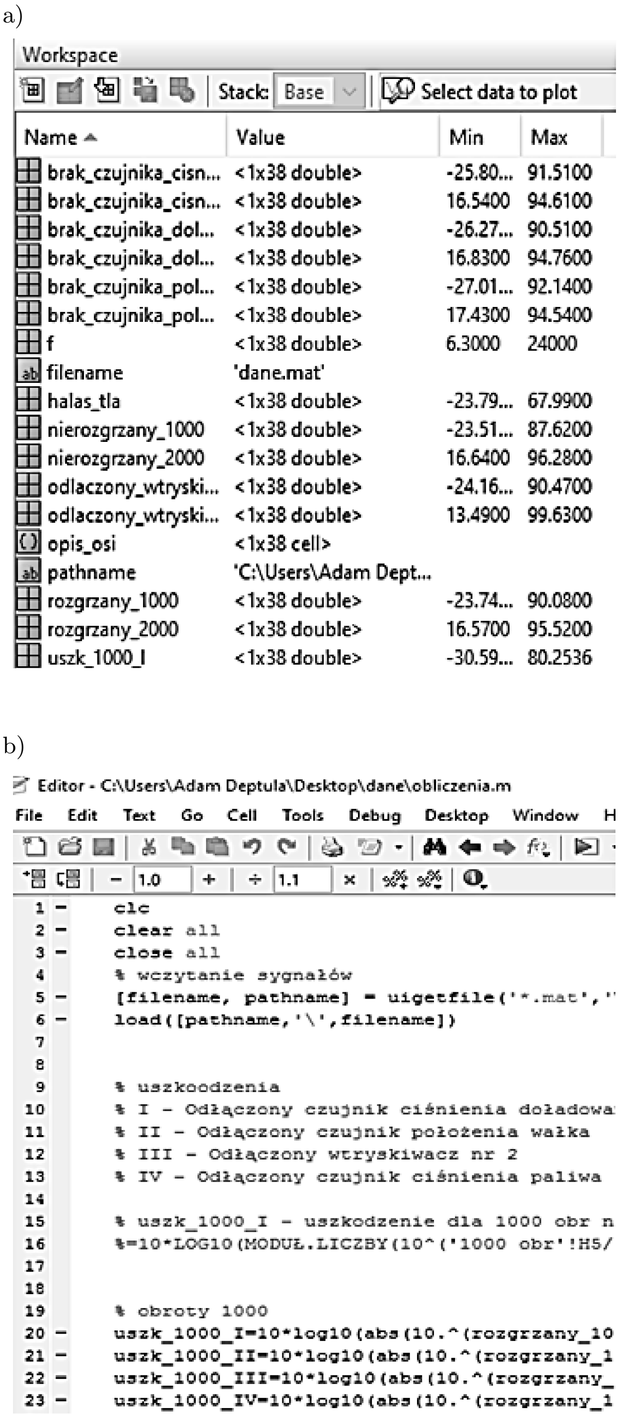


Fig. 14. Identification algorithm engine damage.

In the first stage, data is generated for Matlab, where the file is created dane.mat. Then it creates a morphological matrix (Fig. 15).

For the analysis of graphs playmakers parametrically used original program and Gephi program. Gephi is an open-source software for visualizing and analyzing large networks graphs. Gephi uses a 3D render engine to display graphs in real-time and speed up the exploration. In the first stage, the program reads data using DataLaboratory (Fig. 16).



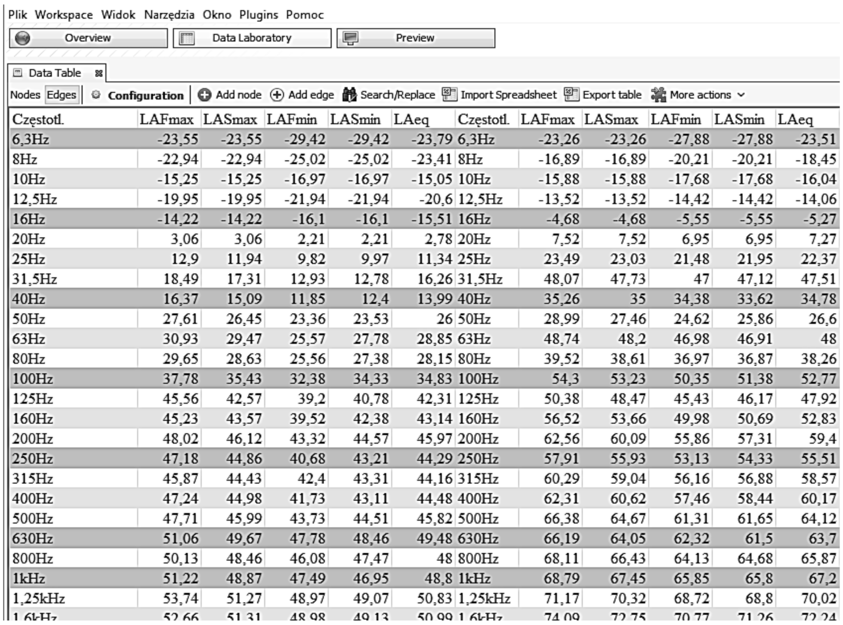
Name ▲	Value	Min	Max
brak_czujnika_cisn...	<1x38 double>	-25.80...	91.5100
brak_czujnika_cisn...	<1x38 double>	16.5400	94.6100
brak_czujnika_dol...	<1x38 double>	-26.27...	90.5100
brak_czujnika_dol...	<1x38 double>	16.8300	94.7600
brak_czujnika_pol...	<1x38 double>	-27.01...	92.1400
brak_czujnika_pol...	<1x38 double>	17.4300	94.5400
f	<1x38 double>	6.3000	24000
filename	'dane.mat'		
halas_tla	<1x38 double>	-23.79...	67.9900
nierozgrzany_1000	<1x38 double>	-23.51...	87.6200
nierozgrzany_2000	<1x38 double>	16.6400	96.2800
odlaczony_wtryski...	<1x38 double>	-24.16...	90.4700
odlaczony_wtryski...	<1x38 double>	13.4900	99.6300
opis_osi	<1x38 cell>		
pathname	'C:\Users\Adam Dept...		
rozgrzany_1000	<1x38 double>	-23.74...	90.0800
rozgrzany_2000	<1x38 double>	16.5700	95.5200
uszk_1000_I	<1x38 double>	-30.59...	80.2536

```
clc
clear all
close all
% wczytanie sygnałów
[filename, pathname] = uigetfile('*.mat','
load([pathname,'\',filename])


% uszkoodzenia
% I - Odłączony czujnik ciśnienia doładowa
% II - Odłączony czujnik położenia wałka
% III - Odłączony wtryskiwacz nr 2
% IV - Odłączony czujnik ciśnienia paliwa

% uszk_1000_I - uszkodzenie dla 1000 obr n
%=10*LOG10(MODUŁ.LICZBY(10^('1000 obr'!H5/


% obroty 1000
uszk_1000_I=10*log10(abs(10.^(rozgrzany_10
uszk_1000_II=10*log10(abs(10.^(rozgrzany_1
uszk_1000_III=10*log10(abs(10.^(rozgrzany_
uszk_1000_IV=10*log10(abs(10.^(rozgrzany_1
```

Fig. 15. Fragments of the source files: a) dane.mat, b) obliczenia.m.

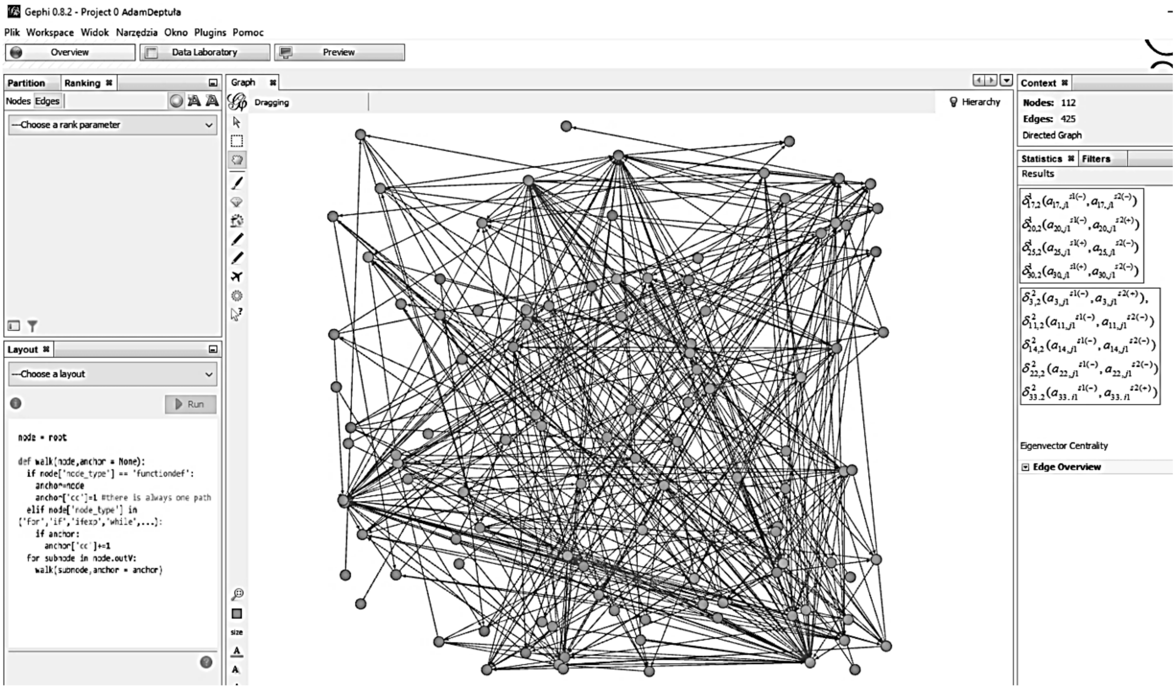
Next it generates a game graph G as a “calculation unit” (Fig. 17).

The sets of highlighted values presenting features of sound pressure values for a given damages have been obtained as a result of an analysis. The above summaries make it possible to create a measuring ladder, which will make it possible to determine the type of engine damage on the basis of sound pressure measurements on chosen frequencies.



Częstotl.	LAFmax	LASmax	LAFmin	LASmin	LAeq	Częstotl.	LAFmax	LASmax	LAFmin	LASmin	LAeq
6,3Hz	-23,55	-23,55	-29,42	-29,42	-23,79	6,3Hz	-23,26	-23,26	-27,88	-27,88	-23,51
8Hz	-22,94	-22,94	-25,02	-25,02	-23,41	8Hz	-16,89	-16,89	-20,21	-20,21	-18,45
10Hz	-15,25	-15,25	-16,97	-16,97	-15,05	10Hz	-15,88	-15,88	-17,68	-17,68	-16,04
12,5Hz	-19,95	-19,95	-21,94	-21,94	-20,6	12,5Hz	-13,52	-13,52	-14,42	-14,42	-14,06
16Hz	-14,22	-14,22	-16,1	-16,1	-15,51	16Hz	-4,68	-4,68	-5,55	-5,55	-5,27
20Hz	3,06	3,06	2,21	2,21	2,78	20Hz	7,52	7,52	6,95	6,95	7,27
25Hz	12,9	11,94	9,82	9,97	11,34	25Hz	23,49	23,03	21,48	21,95	22,37
31,5Hz	18,49	17,31	12,93	12,78	16,26	31,5Hz	48,07	47,73	47	47,12	47,51
40Hz	16,37	15,09	11,85	12,4	13,99	40Hz	35,26	35	34,38	33,62	34,78
50Hz	27,61	26,45	23,36	23,53	26	50Hz	28,99	27,46	24,62	25,86	26,6
63Hz	30,93	29,47	25,57	27,78	28,85	63Hz	48,74	48,2	46,98	46,91	48
80Hz	29,65	28,63	25,56	27,38	28,15	80Hz	39,52	38,61	36,97	36,87	38,26
100Hz	37,78	35,43	32,38	34,33	34,83	100Hz	54,3	53,23	50,35	51,38	52,77
125Hz	45,56	42,57	39,2	40,78	42,31	125Hz	50,38	48,47	45,43	46,17	47,92
160Hz	45,23	43,57	39,52	42,38	43,14	160Hz	56,52	53,66	49,98	50,69	52,83
200Hz	48,02	46,12	43,32	44,57	45,97	200Hz	62,56	60,09	55,86	57,31	59,4
250Hz	47,18	44,86	40,68	43,21	44,29	250Hz	57,91	55,93	53,13	54,33	55,51
315Hz	45,87	44,43	42,4	43,31	44,16	315Hz	60,29	59,04	56,16	56,88	58,57
400Hz	47,24	44,98	41,73	43,11	44,48	400Hz	62,31	60,62	57,46	58,44	60,17
500Hz	47,71	45,99	43,73	44,51	45,82	500Hz	66,38	64,67	61,31	61,65	64,12
630Hz	51,06	49,67	47,78	48,46	49,48	630Hz	66,19	64,05	62,32	61,5	63,7
800Hz	50,13	48,46	46,08	47,47	48	800Hz	68,11	66,43	64,13	64,68	65,87
1kHz	51,22	48,87	47,49	46,95	48,8	1kHz	68,79	67,45	65,85	65,8	67,2
1,25kHz	53,74	51,27	48,97	49,07	50,83	1,25kHz	71,17	70,32	68,72	68,8	70,02
1,6kHz	52,66	51,31	48,98	49,13	50,99	1,6kHz	74,09	72,75	70,77	71,26	72,24

Fig. 16. Loading a matrix of morphological – DataLaboratory.



```
node = root

def walk(node,anchor = None):
  if node['node_type'] == 'functiondef':
    anchor=node
    anchor['cc']=1 #there is always one path
  elif node['node_type'] in
('for','if','ifexp','while',...):
    if anchor:
      anchor['cc']+=1
  for subnode in node.outV:
    walk(subnode,anchor = anchor)
```

Nodes: 112
Edges: 425
Directed Graph

$\delta^1_{17,2}(a_{17,j1}{}^{s1(-)}, a_{17,j1}{}^{s2(-)})$
$\delta^1_{20,2}(a_{20,j1}{}^{s1(-)}, a_{20,j1}{}^{s2(+)})$
$\delta^1_{25,2}(a_{25,j1}{}^{s1(+)}, a_{25,j1}{}^{s2(-)})$
$\delta^1_{30,2}(a_{30,j1}{}^{s1(+)}, a_{30,j1}{}^{s2(-)})$

$\delta^2_{3,2}(a_{3,j1}{}^{s1(-)}, a_{3,j1}{}^{s2(+)})$,
$\delta^2_{11,2}(a_{11,j1}{}^{s1(-)}, a_{11,j1}{}^{s2(-)})$
$\delta^2_{14,2}(a_{14,j1}{}^{s1(-)}, a_{14,j1}{}^{s2(-)})$
$\delta^2_{22,2}(a_{22,j1}{}^{s1(-)}, a_{22,j1}{}^{s2(-)})$
$\delta^2_{33.2}(a_{33.j1}{}^{s1(-)}, a_{33.j1}{}^{s2(+)})$

Fig. 17. Interface user computing – game graph.

6. Conclusions

The authors proposed in the paper decision method for identifying engine failure based on sound emission spectral characteristics. It allows fast identification of specific engine damage. The complete procedure of fault engine diagnostics proposed by authors include several steps. Firstly, the precise sound pressure level characteristics of operating engine need to be obtain for efficient and damaged engine in order to gain engine acoustic maps. The next step is to implement a graph based on the measurement results collected in the form of table. In the decision algorithm, the results are red directly from the prepared table. The game graph, taking into account the search algorithm, finds a characteristic features, points and values (individual for each engine damage) by which the acoustical spectrum may be assigned to specified type of damage. The method based on game graphs allows to determine technical condition of engine using a small number of measurements. In the paper, a complete program for decision support system was shown, which may be successfully used for subsequent measurements on the same engine and engines of the same kind.

References

1. Adaileh W.M. (2013), *Engine Fault Diagnosis Using Acoustic Signals*, Applied Mechanics and Materials, **295–298**, 2013–2020.
2. Barelli L., Bidini G., Buratti C., Mariani R. (2009), *Diagnosis of internal combustion engine through vibration and acoustic pressure non-intrusive measurements*, Applied Thermal Engineering, **29**, 1707–1713.
3. Carlucci A.P., Chiara F.F., Laforgia D. (2006), *Analysis of the relation between injection parameter variation and block vibration of an internal combustion diesel engine*, Journal of Sound and Vibration, **295**, 141–164.
4. Deptula A. (2015), *Application of game graphs to describe the inverse problem in the designing of mechatronic vibrating systems*, [in:] Graph-Based Modelling in Engineering, Zawiślak S., Rysiński J. [Eds.], Springer [in print].
5. Deptula A. (2014), *Application of multi-valued weighting logical functions in the analysis of a degree of importance of construction parameters on the example of hydraulic valves*, International Journal of Applied Mechanics and Engineering, University Press Zielona Góra, **19**, 3, 539–548, ISSN 1425-16554.
6. Deptula A., Partyka M.A. (2011), *Application of dependence graphs and game trees for decision decomposition for machine systems*, Journal of Automation, Mobile Robotics & Intelligent Systems, **5**, 3, 17–26.
7. Deptula A., Partyka M.A. (2014), *Decision optimization of machine sets with taking into consideration logical tree minimization of design guidelines*, International Journal of Applied Mechanics and Engineering, University Press Zielona Góra, **19**, 3, 549–561, ISSN 1425-16554.
8. Divina F., Marchiori E. (2005), *Handling Continuous Attributes in an Evolutionary Inductive Learner*, IEEE Transactions on Evolutionary Computation, **9**, 1, 31–43.
9. Elamin F., Gu F., Ball A. (2010), *Diesel Engine Injector Faults Detection Using Acoustic Emissions Technique*, Modern Applied Science, **4**, 9.
10. Ettefagh M.M., Sadeghi M.H., Pirouzpanah V., Arjmandi Tash H. (2008), *Knock detection in spark ignition engines by vibration analysis of cylinder block: A parametric modeling approach*, Mechanical Systems and Signal Processing, **22**, 1495–1514.
11. Głowacz A. (2014), *Diagnostics of Synchronous Motor Based on Analysis of Acoustic Signals with the use of Line Spectral Frequencies and K-nearest Neighbor Classifier*, Archives of Acoustics, **39**, 2, 189–194.
12. Głowacz A. (2015), *Recognition of acoustic signals of synchronous motors with the use of MoFS and selected classifiers*, Measurement Science Review, **15**, 4, 167–175.
13. Głowacz A., Głowacz W. (2010), *Diagnostics of synchronous motor based on analysis of acoustic signals with application of LPCC and nearest mean classifier with cosine distance*, Archives of Metallurgy and Materials, **55**, 2, 563–570.
14. Głowacz Z., Kozik J. (2013), *Detection of synchronous motor inter-turn faults based on spectral analysis of park's vector*, Archives of Metallurgy and Materials, **58**, 1, 19–23.
15. Kirpluk M. (2012), *Fundamentals of acoustics* [in Polish], Wyd. NTL-M, Warszawa.
16. Luft S. (2010), *Basics of Engines Construction* [in Polish], Wyd. WKŁ, Warszawa.
17. Osiński P., Deptuła A., Partyka M.A. (2013), *Discrete optimization of a gear pump after tooth root undercutting by means of multi-valued logic trees*, Archives of Civil and Mechanical Engineering, **13**, 4, 422–431.
18. Osiński P., Kollek W. (2013), *Assessment of energetistic measuring techniques and their application to diagnosis of acoustic condition of hydraulic machinery and equipment*, Archives of Civil and Mechanical Engineering, **13**, 3, 313–321, http://dx.doi.org/10.1016/j.acme.2013.03.001.
19. Ranachowski Z., Bejger A. (2005), *Fault diagnostics of the fuel injection system of a medium power maritime diesel engine with application of acoustic signal*, Archives of Acoustics, **30**, 4, 465–472.
20. Serdecki W. (2012), *Combustion engines testing* [in Polish], Wydawnictwo Politechniki Poznańskiej, Poznań.
21. Sobański P., Orłowska-Kowalska T. (2015), *Faults diagnosis and control in a low-cost fault-tolerant induction motor drive system*, Mathematics and Computers in Simulation, http://dx.doi.org/10.1016/j.matcom.2015.10.012.
22. Teodorczyk A., Rychter T. (2010), *Piston Engine theory* [in Polish], Wyd. WKŁ, Warszawa.
23. Wu J.D., Chen J.Ch. (2006), *Continuous wavelet transform technique for fault signal diagnosis of internal combustion engines*, NDT&E International, **39**, 304–311.
24. Wu J.D., Kuo J.M. (2009), *An automotive generator fault diagnosis system using discrete wavelet transform and artificial neural network*, Expert Systems with Applications, 36, 9776–9783.
25. Wu J.D., Liu Ch.H. (2009), *An expert system for fault diagnosis in internal combustion engines using wavelet packet transform and neural network*, Expert Systems with Applications, 36, 4278–4286.
26. Zhen D., Wang T., Gu F., Tesfa B., Ball A. (2013), *Acoustic measurements for the combustion diagnosis of diesel engine fuelled with biodiesels*, Meas. Sci. Technol., 24, 13.