SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 20, No. 1, February 2023, 93-105
UDC: 004.891.3:621.3.067 **DOI: https://doi.org/10.2298/SJEE2301093A**

Expert System for FDI of DC Motor Faults Using Structured Residuals Design Technique

Sanja Antić[1], Vanja Luković[1], Željko Đurović[2]

Abstract: A major concern in many electrical drives is the reliability of sensors and actuators. In the paper, the usage of the Drools expert system (ES) for Fault detection and isolation (FDI) of the additive actuator and sensor DC Motor faults using the Structured residuals design technique (SRDT) is presented. The SRDT is used to obtain essential knowledge about the system. Afterward, an expert system that can isolate faults based on the developed structure matrix and generated residuals is designed. Accordingly, following the structure matrix each residual becomes able to answer to a desired subset of faults and stands insensitive to the others. The proposed method is successfully applied in an analyzed laboratory system and can be used for online FDI.

Keywords: Expert Systems, Drools, Fault detection and isolation, Structured residuals design technique, Parity equations, DC motor.

1 Introduction

Electric motors are widely used in industry. Accordingly, FDI on these electro-mechanical components and their actuators and sensors is of great importance for improving the availability, reliability, and security of the entire system. Analyzing techniques applied to motors, sensors, and actuators, three specific approaches can be distinguished:

1. Methods based on vibration analysis, analysis of prints, or signal models [1 – 3]. The advantage of these methods is that they do not require precise modeling of the system. However, they are not suitable for fast online testing.
2. Methods based on the dynamic model of the system [4 – 7]. The approach based on the formation of parity equations and structural residuals design technique was developed by Gertler [8 – 10] and represents an efficient method in fault isolation.

[1]Faculty of Technical Sciences Cacak, Svetog save 65, University of Kragujevac, 32000 Serbia; E-mails: sanja.antic@ftn.kg.ac.rs, vanja.lukovic@ftn.kg.ac.rs
[2]School of Electrical Engineering Belgrade, Bulevar kralja Aleksandra 73, University of Belgrade, 11000, Serbia; E-mail: zeljko.djurovic@etf.bg.ac.rs

3. Methods based on experience and knowledge which include the synthesis of neural networks, the application of methods of expert systems, and fuzzy logic [11–14]. The problem with the use of these methods is that they are time-consuming and difficult to apply since they involve accumulating experience and expressing it through appropriate laws.

D. Fuessel and R. Isermann propose a fuzzy logic scheme for fault diagnosis on the example of a direct current motor [11]. The advantage of the method is increased robustness compared to traditional classification schemes. K. Patan analazed robust diagnosis of DC motor faults using artificial neural networks and model error modeling [12].

An ES for fault diagnosis, repair, and maintenance of electrical machines using vibration signal analysis is presented in [13]. Empirical rules are used to help the user to find and repair any known fault in any electrical machine. F. Filippetti et al. highlight that an ES can be built to detect and isolate sensor and actuator motor faults according to the experience accumulated by an experienced engineer, by observing the motor operating behavior [14]. The time-consuming problem and accumulating experience are also present here. But once the knowledge base (KB) is created, ES can be successfully used for FDI.

This paper presents sensor and actuator FDI of DC motor additive faults. Two approaches were applied: a model-based approach based on parity equations with structural residual design and a knowledge-based ES method. The structural residual technique (SRDT) was used to collect necessary knowledge about the system. Then, based on the collected information, and stored residuals, a knowledge-based ES method is applied to isolate all of the faults [15].

The proposed approach can be used for online FDI and students training in the field of FDI.

2 Equipment and Models

The analyzed fault detection system consists of a permanent-magnet (PM) DC micro-motor, an amplifier formed as a linear electronic circuit, and CRIO 9075 controller. Fig. 1. The DC motor and amplifier data are shown in **Table 1**.

A compact CRIO 9075 integrated system [16] combines a 400-MHz industrial real-time processor with LX25 FPGA and has four slots for NI C Series I/O modules. It is suitable for motor control and monitoring applications. The used modules in the experiment were: NI 9402 for speed measurement, NI 9205 for motor voltage and current measurement, and NI 9263 for assigning the control voltage to the amplifier.

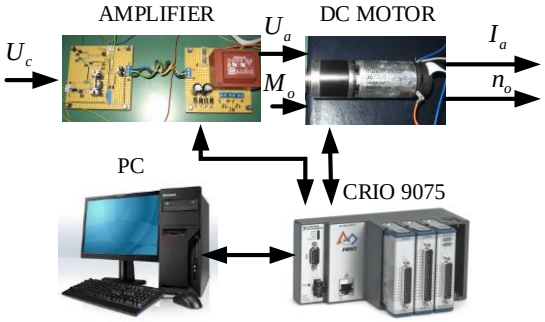


Fig. 1 – *Structural components of the system.*

Table 1

A-max26 ø26-mm, 4.5W - PM DC motor, and amplifier data.

SYMBOLS	NAME	VALUES
U_n	Nominal voltage	12 [V]
I_n	Nominal current	0.629[A]
n_n	Nominal speed	2750 [rpm]
K_T	Torque constant	$25.5 \cdot 10^{-3}$ [Nm/A]
K_e	Back ems constant	$25.5 \cdot 10^{-3}$ [Nm/rad/s]
R_a	Armature resistance	7.41 [Ω]
L_a	Armature inductance	$0.77 \cdot 10^{-3}$[H]
J	Motor moment of inertia	$1.43 \cdot 10^{-3}$ [kgm^2]
N	Reductor gain	3.8
R_{1e}	Amplifier equivalent resistance 1	5 [kΩ]
R_{2e}	Amplifier equivalent resistance 2	12 [kΩ]
R_1	Amplifier resistance	1[Ω]

In conditions of absence of faults, choosing a sampling time of T= 10ms which is in accordance with the compensation theorem, Z-transform discretized transfer functions of the system were determined (1) - (5).

$$M_e(z) = \frac{U_a(k)}{U_c(k)} = \frac{2.115 - 1.098z^{-1}}{1 - 0.5324z^{-1}}, \tag{1}$$

$$M_{11}(z) = \frac{I_a(k)}{U_a(k)} = \frac{0.07406z^{-1} - 0.07279z^{-2}}{1 - 0.5324z^{-1}}, \tag{2}$$

$$M_{12}(z) = \frac{I_a(k)}{M_o(k)} = \frac{17.83z^{-1} + 0.1358z^{-2}}{1 - 0.5324z^{-1}}, \tag{3}$$

$$M_{21}(z)=\frac{n_o(k)}{U_a(k)}=\frac{44.81z^{-1}+0.3412z^{-2}}{1-0.5324z^{-1}}, \tag{4}$$

$$M_{22}(z)=\frac{n_o(k)}{M_o(k)}=\frac{-1.312\cdot10^{4}z^{-1}-0.6364z^{-2}}{1-0.5324z^{-1}}, \tag{5}$$

where the inputs of the system are: the control voltage $U_c(k)$ and load torque $M_o(k)$, while the outputs are: the armature voltage $U_a(k)$ armature current $I_a(k)$, and motor speed after the gear $n_o(k)$.

3 Fault Detection and Isolation Using SRDT

Five different DC motor additive faults were considered in the research: two of them are input sensor faults (during control signal generation $\Delta U_c(k)$ and torque measurement $\Delta M_o(k)$) and three of them are output sensor faults (during measurement of armature voltage $\Delta U_a(k)$, armature current $\Delta I_a(k)$, and motor speed $\Delta n_o(k)$). The steps in designing structural residual generator are:

1. The system modeling in the presence of faults, according to:

$$\boldsymbol{y}(k)=\boldsymbol{M}(z)\boldsymbol{u}(k)+\boldsymbol{S}_F(z)\boldsymbol{p}(k), \tag{6}$$

where $\boldsymbol{M}(z)$ is the system transfer function matrix and $\boldsymbol{S}_F(z)$ is the fault transfer function matrix:

$$\boldsymbol{M}(z)=\begin{bmatrix} M_e(z) & 0 \\ M_{11}(z)M_e(z) & M_{12}(z) \\ M_{21}(z)M_e(z) & M_{22}(z) \end{bmatrix},$$
$$\boldsymbol{S}_F(z)=\begin{bmatrix} M_e(z) & 0 & 1 & 0 & 0 \\ M_{11}(z)M_e(z) & M_{12}(z) & 0 & 1 & 0 \\ M_{21}(z)M_e(z) & M_{22}(z) & 0 & 0 & 1 \end{bmatrix}, \tag{7}$$

and $\boldsymbol{p}(k)$, $\boldsymbol{u}(k)$, $\boldsymbol{y}(k)$ are the fault, input, and output vectors:

$$\boldsymbol{p}(k)=\begin{bmatrix}\Delta U_c(k) & -\Delta M_o(k) & \Delta U_a(k) & \Delta I_a(k) & \Delta n_o(k)\end{bmatrix}', \tag{8}$$

$$\boldsymbol{u}(k)=[U_c(k) \quad M_o(k)]', \tag{9}$$

$$\boldsymbol{y}(k)=[U_a(k) \quad I_a(k) \quad n_o(k)]'. \tag{10}$$

2. Derivation of the internal primary residuals using the equation:

$$\boldsymbol{o}(k)=\boldsymbol{S}_F(\mathrm{z})\boldsymbol{p}(k). \tag{11}$$

From (7) and (11), the internal primary residuals defined from the transfer function model are:

$$o_1 = M_e(z)\Delta U_c + \Delta U_a, \tag{12}$$

$$o_2 = M_{11}(z)M_e(z)\Delta U_c + M_{12}(z)(-\Delta M_o) + \Delta I_a, \tag{13}$$

$$o_3 = M_{21}(z)M_e(z)\Delta U_c + M_{22}(z)(-\Delta M_o) + \Delta n_o. \tag{14}$$

3. Structural matrix design. From the internal primary residual set one of the possible structural matrices has the form presented in (15), [15]

$$\begin{array}{c|ccccc} & p_1 & p_2 & p_3 & p_4 & p_5 \\ r_1 & 1 & 1 & 0 & 1 & 0 \\ r_2 & 1 & 1 & 0 & 0 & 1 \\ r_3 & 1 & 0 & 1 & 0 & 0 \\ r_4 & 0 & 0 & 1 & 1 & 1 \\ r_5 & 0 & 1 & 1 & 1 & 0 \end{array} \tag{15}$$

4. Synthesis of the transformation matrix $\boldsymbol{W}(z)$ using selected design method [9].
5. Structural residual generation based on the following equation:

$$\boldsymbol{r}(k) = \boldsymbol{W}(z)\boldsymbol{o}(k). \tag{16}$$

4 Expert Systems

Expert systems (ES) are intelligent programs that are used to solve problems in different areas that would otherwise require human expertise. ESs are based on a large amount of high-quality knowledge collected by experts, embedded in a knowledge base (KB) (Fig. 2). KB contains domain knowledge that must be formalized so that the computer can use it. The most commonly used technique for presenting knowledge within the ES is the rule-based technique. The basic and most important characteristic of the rule is that they can be "chained". Working memory (WM) contains facts and conclusions. The conclusions are facts that were created as a result of reasoning. The facts must also be formalized so that the computer can use them. The most common way for presenting facts within the ES is based on objects (classes). The inference engine enables automated reasoning. It combines knowledge from the knowledge base and facts from the working memory and creates new conclusions.

In this paper Drools [17 – 19] (Fig. 3) free, open–source software developed by Knowledge Is Everything (KIE) project was used. It is written in Java programming language and is compatible with all types of operating systems. It integrates into the Eclipse environment in the form of a plug-in. The last stable

version of the Drools Expert tool is 8.32.0. It enables the development, running and testing of all types of ES.

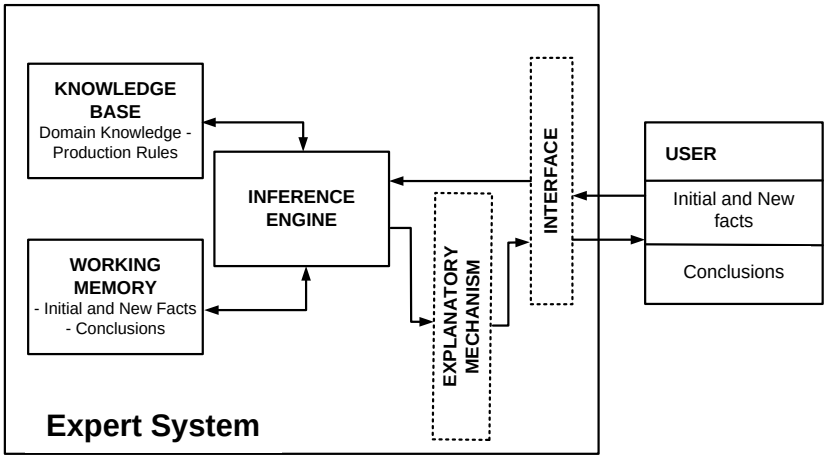


Fig. 2 – *Structural components of an ES.*

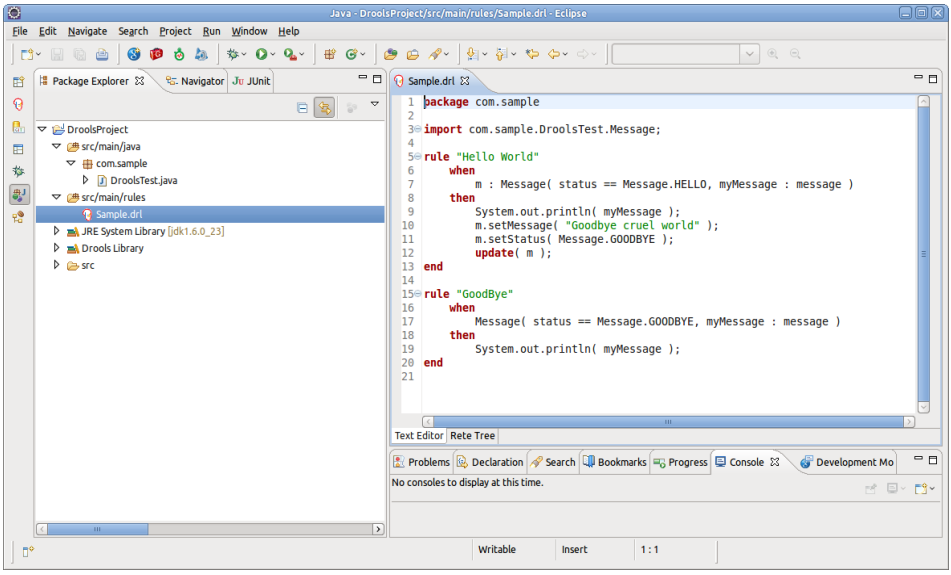


Fig. 3 – *Drools software Expert tool for creating ESs integrated into the Eclipse environment.*

The facts that the ES takes from the user are stored in the Java objects. For Drools to automatically retrieve data from an object, objects must satisfy the JavaBeans standard, i.e. to have: get methods for setting the value of each attribute and set methods for downloading the value of each attribute.

The KB in Drools is formalized using the rules technique and stored in DRL files. The reserved word package allows the division of rules into groups in case the domain problem is complex. The rule syntax is following:

```
rule "title of the rule"
// attributes of the rule
when
// condition (LHS - Left Hand Side)
then
// conclusion (RHS - Right Hand Side)
end
```

The name of the rule can be any string that should be quoted. The attributes are optional and affect the behavior of the rule (priority, validity period, etc.). The rule condition (LHS) contains one or more logical statements that resemble logical expressions in Java. It expresses constraints on the data model using the class name and its attributes. The conclusion of the rule (RHS) is given in the form of ordinary Java commands. It acts on the facts that are in the working memory. Drools tool uses a forward chaining technique for reasoning.

Generating the explanation of how the ES concludes can be done using the JEFF (Java Explanation Facility Framework) Wizard [20]. It's a free, open-source Java framework, which explains how an ES concludes. The explanation file can be generated in PDF, XML, or TXT format.

5 Results and Discussion

On the basis of the generated structural matrix (15), the Drools ES for the detection and isolation of additive DC motor faults was formed. The system is also applicable for online fault detection. Faults were individually, and artificially entered into the system. Afterward, the computational primary residuals were calculated based on the principle of Analytical redundancy. The measured output was compared with the output obtained from the system model, according to the following equation:

$$\boldsymbol{o}(k) = \boldsymbol{y}(k) - \boldsymbol{M}(\mathrm{z})\boldsymbol{u}(k). \tag{17}$$

Then, the structured residuals were calculated according to (16), filtered and stored as text files. The obtained residuals are shown in Figs. 4 – 8, [15].

To detect the presence of a fault which means to indicate the presence of a fault, it is usually enough that any residual becomes greater or equal then the predefined threshold value 1. But for fault isolation, i.e. to determine where is the fault, only a specific set of residuals must respond according to the structure matrix (15). So, for example, at the occurrence of the first fault $p_1 = \Delta U_c$, to isolate this fault, residuals r_1, r_2, and r_3 in the stationary state must become greater or equal to a pre-defined threshold value 1.

These defined rules are met by all residuals, which confirms that all of the faults can be isolated (Figs. 4 – 8).

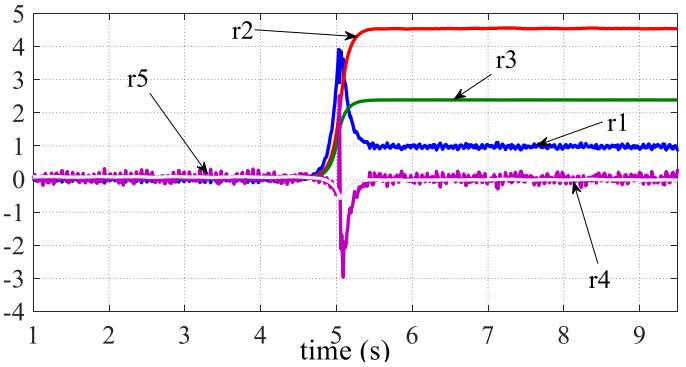


Fig. 4 – *Isolation of the fault* $p_1 = \Delta U_c$.

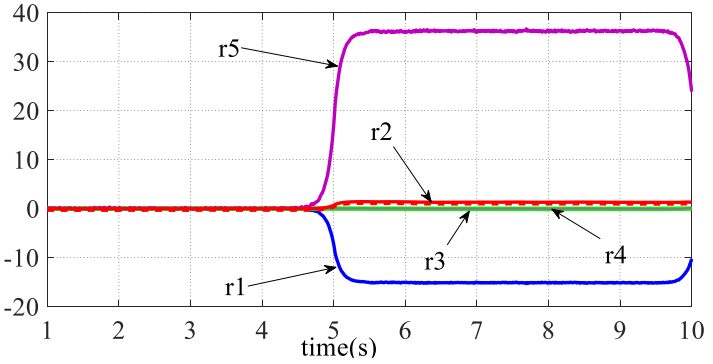


Fig. 5 – *Isolation of the fault* $p_2 = -\Delta M_o$.

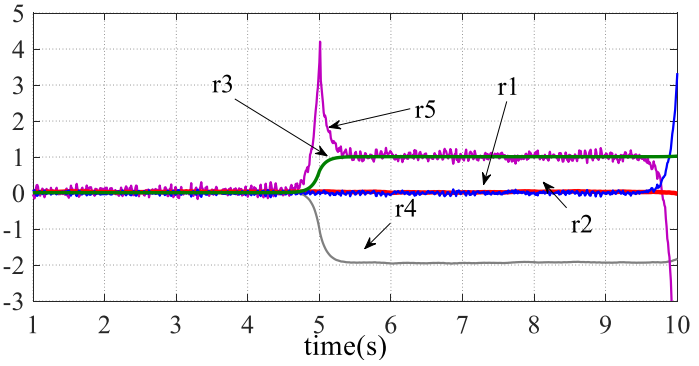


Fig. 6 – *Isolation of the fault* $p_3 = \Delta U_a$.

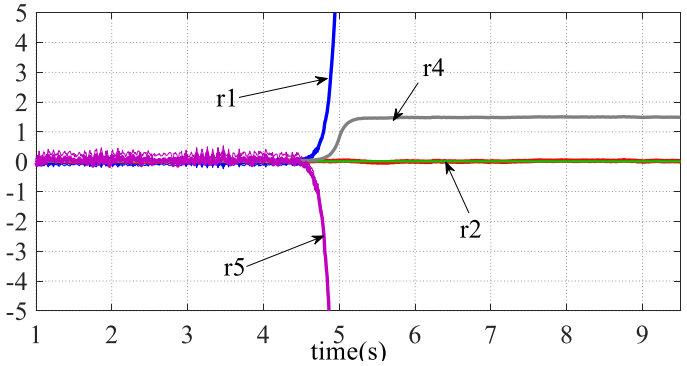


Fig. 7 – *Isolation of the fault* $p_4 = \Delta I_a$.

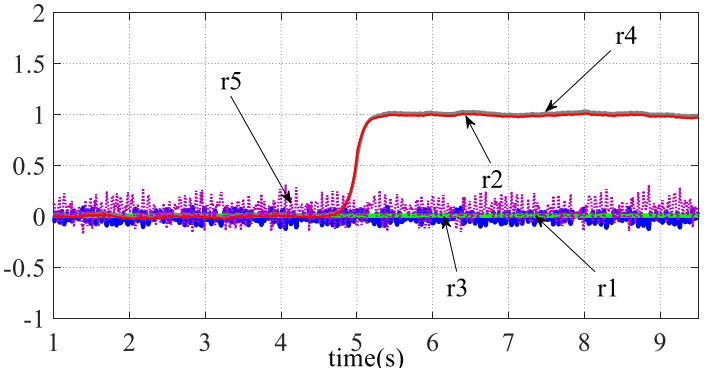


Fig. 8 – *Isolation of the fault* $p_5 = \Delta n$.

To create the ES for detecting DC motor faults a Java class named *Residuals*, with attributes representing residuals values (r_1, r_2, r_3, r_4, and r_5), fault time (*time*), and fault duration (*duration*) was defined. The class also contains attributes describing the fault type (*FalutIsolation*) and fault identification (*FaultIdentification*) and get and set methods for getting and setting attribute values.

Also, the knowledge base that consists of a set of rules for FDI was created. The rules were stored in a separate DRL file named *rules.drl*. Each DRL file must belong to exactly one package, in this case, to package named *rules*. The reserved import word serves to import all Java classes that will be used within the rules.

In our case rule conditions (LHS) check residual attribute values. The conclusion of the rule (RHS) is given in the form of ordinary Java commands. It acts on the facts that are in the working memory. In our case, it sets values of attributes describing fault type and fault isolation and generates an explanation of

how the ES came to a conclusion using the *JEFFWizard* class object, named *ef*. The part of the DRL file is as follows:

```
package rules;
import rs.DroolsTest;
import org.goodoldai.jeff.wizard.JEFFWizard;
import org.goodoldai.jeff.explanation.ImageData;
import rs.Reziduals;
global JEFFWizard ef;
//list any import classes here.

//declare any global variables here

rule "Input actuator fault during control signal generation"
 lock-on-active true
  when
   r : Reziduals(r1>=1, r2>=1, r3>=1, r4==0, r5==0)
  then
   r.setFaultIsolation("Attention! Input actuator fault during
control signal generation.");
   r.setFaultIdentification("Check the power supply of the
amplifier!");
   update(r);

ef.addText(null,"================================================
==========================");
   ef.addText(null, "Fault type: Input actuator fault during
control signal generation");
   ef.addText(null, "Fault time: " + r.getTime() + "ms");
   ef.addText(null, "Fault duration: " + r.getDuration() + "ms");
   ef.addText(null, "Solution: Check the power supply of the
amplifier!\n");
   ef.addImage(new ImageData("/Motor1r.jpg"));

end
```

Running the ES is done in the main function of the application, which is in our case located in the *DroolsTest.java* file, using the following steps:

1. Reading the Knowledge Base from the DRL file is done using *RuleBase* class and *PackageBuilder* classes;
2. Creation of facts (Java objects) - in our case, an array of *Residuals* objects is created by reading the text file *Residuals.txt* that contains the residual values, which are generated every 10s by the system and recorded in an array element;
3. Inserting facts into working memory (adding objects to working memory) is done using the *WorkingMemory* class and *insert*() method, every 10s;
4. Running the ES is done using the *WorkingMemory* class and the *fireAllRules*() method;

5. Generating the explanation of how the ES came to a conclusion is done using *JEFFWizard* class with the possibility of inserting the images (*addImage*() method) and text (*addText*() method). In our case, the PDF format of the explanation file is used (*generatePDFReport*() method).

An example of the generated report is given in Fig. 9.

DC motor fault detection and isolation

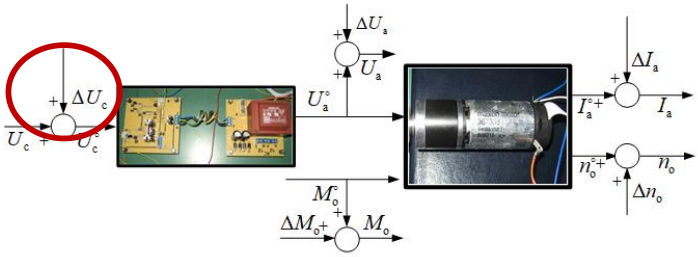
==========

Fault type: Input actuator fault during control signal generation

Fault time: at 5s

Fault duration: 5s

Solution: Check the power supply of the amplifier



	p_1	p_2	p_3	p_4	p_5
r_1	1	1	0	1	0
r_2	1	1	0	0	1
r_3	1	0	1	0	0
r_4	0	0	1	1	1
r_5	0	1	1	1	0

Fig. 9 – *Part of the explanation file.*

6 Conclusion

The paper presents the application of a combination of a method based on system modeling using the Structured residuals design technique and a method that uses knowledge-based expert system. Primarily, the parity equations method based on the SRDT was used to acquire the necessary knowledge about the system, generate a structural matrix, and form residuals. Then, the ES was designed for fault isolation.

The entire FDI system can also be applied online, which is the advantage of our proposed approach. This is realized based on the principle of Analytical redundancy. That implies the measured output is compared with the output obtained from the system model to form primary residuals. Then, by multiplying these computational primary residuals with the already synthesized transformation matrix, the residuals are formed. They are then filtered, and perhaps

additionally processed, saved as text files, and introduced into the designed Expert system to isolate faults in real-time.

The presented method is applied to a laboratory system that is often encountered in many laboratories and can be educationally convenient and useful for students training in the field of FDI. The proposed FDI approach can also be applied to any control system that can be modeled. In the case of industrial application, instead of modeling, a more suitable approach would be system identification and then the design of an Expert system. Also, the application of some other FDI approaches such as methods based on the signal model and signal prints analysis, observer and filter techniques, or other knowledge-based procedures are all applicable in the industry. This would be our future direction of research.

7 Acknowledgments

This study was supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia, and these results are parts of the Grant No. 451-03-47/2023-01/200132 with University of Kragujevac - Faculty of Technical Sciences Čačak.

8 References

[1] W. Saadaoui, K. Jelassi: Induction Motor Bearing Damage Detection Using Stator Current Analysis, Proceedings of the International Conference on Power Engineering, Energy and Electrical Drives, Málaga, Spain, May 2011, pp. 1 – 6.

[2] W. Li, J. Fang, H. Li: Single Switch Open-Circuit Fault Diagnosis in Brushless DC Motor Drivers with Buck Converters, Proceedings of the Chinese Society for Electrical Engineering, Vol. 33, No. 15, 2013, pp. 124 – 132.

[3] S. Rajagopalan, J.M. Aller, J.A. Restrepo, T.G. Habetler, R.G. Harley: Detection of Rotor Faults in Brushless DC Motors Operating Under Nonstationary Conditions, IEEE Transactions on Industry Applications, Vol. 42, No. 6, November 2006, pp. 1464 – 1477.

[4] R. Isermann: Model-Based Fault-Detection and Diagnosis – Status and Applications, Annual Reviews in Control, Vol. 29, No. 1, 2005, pp. 71 – 85.

[5] M. El-Koujok, M. Benammar, N. Meskin, M. Al-Naemi, R. Langari: Multiple Sensor Fault Diagnosis by Evolving Data-Driven Approach, Information Sciences, Vol. 259, February 2014, pp. 346 – 358.

[6] W. Li, S. Shah: Structured Residual Vector-Based Approach to Sensor Fault Detection and Isolation, Journal of Process Control, Vol. 12, No. 3, April 2002, pp. 429 – 443.

[7] A. Asokan, D. Sivakumar: Model Based Fault Detection and Diagnosis Using Structured Residual Approach in a Multi-Input Multi-Output System, Serbian Journal of Electrical Engineering, Vol. 4, No. 2, November 2007, pp. 133 – 145.

[8] J. Cao, J. Gertler: Partial PCA-Based Optimal Structured Residual Design for Fault Isolation, Proceedings of the American Control Conference, Boston, USA, June 2004, pp. 4420 – 4425.

[9] J. J. Gertler: Fault Detection and Diagnosis in Engineering Systems, 1st Edition, CRC Press, Boca Raton, London, New York, 2019.

[10] J. Gertler: Fault Detection and Isolation Using Parity Relations, Control Engineering Practice, Vol. 5, No. 5, May 1997, pp. 653 – 661.

[11] D. Fuessel, R. Isermann: Hierarchical Motor Diagnosis Utilizing Structural Knowledge and a Self-Learning Neuro-Fuzzy Scheme, IEEE Transactions on Industrial Electronics, Vol. 47, No. 5, October 2000, pp. 1070 – 1077.

[12] K. Patan: Robust Faul Diagnosis in a DC Motor by Means of Artificial Neural Networks and Model Error Modelling, Korbicz, J., Patan, K., Kowal, M., eds.: Fault Diagnosis and Fault Tolerant Control, W. Academic Publishing House Exit, Ed. 2007, pp. 337 – 346.

[13] A. Kafeel, S. Aziz, M. Awais, M. A. Khan, K. Afaq, S. A. Idris, H. Alshazly, S. M. Mostafa: An Expert System for Rotating Machine Fault Detection Using Vibration Signal Analysis, Sensors, Vol. 21, No. 22, November 2021, p. 7587.

[14] F. Filippetti, M. Martelli, G. Franceschini, C. Tassoni: Development of Expert System Knowledge base to On-Line Diagnosis of Rotor Electrical Faults of Induction Motors, Proceedings of the Conference Record of the 1992 IEEE Industry Applications Society Annual Meeting, Houston, USA, October 1992, pp. 92 – 99.

[15] S. Antic, Z. Djurovic, G. Kvascev: Application of Structured and Directional Residuals for Fault Detection and Isolation on Permanent-Magnet DC Motor with Amplifier, Quality and Reliability Engineering International, Vol. 32, No. 7, November 2016, pp. 2601 – 2621.

[16] cRIO-9075 - NI, Available at: https://www.ni.com/en-rs/support/model.crio-9075.html (accessed January 12, 2023).

[17] A.Y. Yurin, N.O. Dorodnykh: Personal Knowledge Base Designer: Software for Expert Systems Prototyping, SoftwareX, Vol. 11, January 2020, p. 100411.

[18] R. Müller, L. Hörauf, D. Burkhard: Development of an AI-Based Expert System for the Part- and Process-Specific Marking of Materials, Procedia CIRP, Vol. 100, 2021, pp. 361 – 366.

[19] Drools - Drools - Business Rules Management System (JavaTM, Open Source), Available at: https://www.drools.org/ (accessed January 12, 2023).

[20] JEFF (Java Explanation Facility Framework) – SourceForge open source software-Download, Available at: https://sourceforge.net/projects/jeff/