IOP Conference Series: Materials
Science and Engineering

PAPER • **OPEN ACCESS**

Software developments for an electric motor test bench developed at the Faculty of Engineering of the University of Debrecen

To cite this article: É Ádámkó *et al* 2022 *IOP Conf. Ser.: Mater. Sci. Eng.* **1237** 012012

View the article online for updates and enhancements.

Software developments for an electric motor test bench developed at the Faculty of Engineering of the University of Debrecen

É Ádámkó[1], A Szántó[2] and G Á Sziki[3]

[1,2,3]University of Debrecen, Faculty of Engineering, 4028 Debrecen, Ótemető street 2-4. Hungary

adamko.eva@eng.unideb.hu

Abstract. At the Faculty of Engineering of the University of Debrecen, our research group has long-term experience in the dynamic modeling and simulation of electric motors. Several simulation modules have been developed during the last few years. The input data of the developed simulation modules are the technical parameters and characteristics of the analyzed motors. The mentioned technical data from the manufacturers is either not provided or it's incomplete, thus, it is necessary to measure it. Additionally, the accuracy of the output dynamic functions, generated by the simulation modules, has to be tested experimentally. As a consequence, it was decided by the research group to realize a Measurement System for performing both above-mentioned experimental tasks routinely, at high accuracy and at a precision level. In this paper, a brief description of the Hardware Component of the Measurement System and a full detailed one of the developed Software Component is given. The Software Component of the Measurement System is implemented in NI LabVIEW, Java SE, and VBA with an underlying MySQL database. Applying the Software Component, the measured quantities can be monitored in real-time, displayed graphically while being stored in a database for later use.

1. Introduction

Our research group at the Faculty of Engineering of the University of Debrecen has long-term experience in the modeling and simulation of electric motors and powertrains [1] [2] [3] [4]. To be able to simulate the above motors, their electromagnetic and dynamic characteristics have to be precisely known, since these data serve as input of the simulation program. Most of these data are not given in the catalog – specification – of the motor, so they have to be measured. These missing parameters was the main motivation to develop a complex Measurement System (MS) for the experimental study and testing of electric motors. By applying the MS, all of the electromagnetic and dynamic characteristics of a motor can be measured, additionally, test measurements can be performed on it too [5] [6]. The MS consists of a Hardware Component (HC) and a Software Component (SC). In Figure 1. that particular part of the HC is shown, which is used to perform test measurements on electric motors.

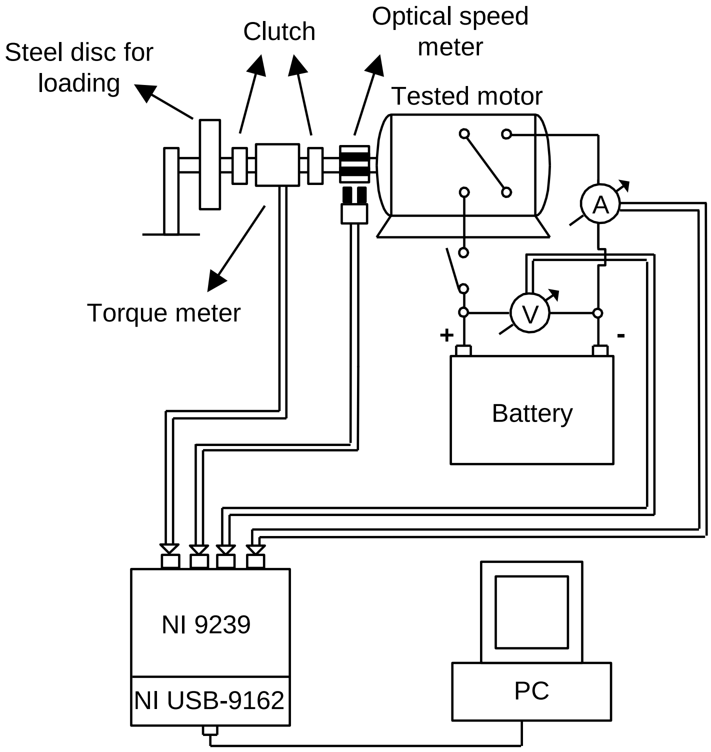


Figure 1. Hardware Component of the Measurement System for test measurements on electric motors

In the course of a typical test measurement, additional loads are applied to the motor with different moment of inertias. As voltage is switched on the motor, its rotor spins up from a state of rest until it reaches the maximum speed possible. During spin up, the angular speed and torque of the motor are measured in real-time. Additionally, the intensity of the electric current flowing through the windings of the motor, and the terminal voltage of the battery is measured, also in real-time. Several sensors are installed into the MS, e.g., optical LED and rotary torque sensors. The output signals of the optical LED and rotary torque sensors – for angular speed and torque measurements –, and also the intensity of the electric current, flowing through the motor, are converted to voltage signals. As it is shown in Figure 1. the output voltage signals of the different sensors and resistors are fed into a(n) *NI 9239* voltage input module, which is connected to a PC through a *CompactDAQ Chassis*. To be able to collect, separate, display, and analyze the digital data transmitted by the connector, a suitable software is necessary. That is the reason for the decision, why a self-developed SC was realized. The SC is primarily implemented in the NI LabVIEW environment [7][8]. In Section 2 the HC is briefly presented, and in Section 3 a detailed description of the SC of the MS is given.

2. Description of the Hardware Component

The wiring diagram of that particular part of the HC which is used during the test measurements on electric motors is shown in Figure 2. As it is mentioned earlier the output voltage signals of the different sensors and resistors are fed into a(n) *NI 9239* voltage input module, which is connected to a PC through a *CompactDAQ Chassis*.

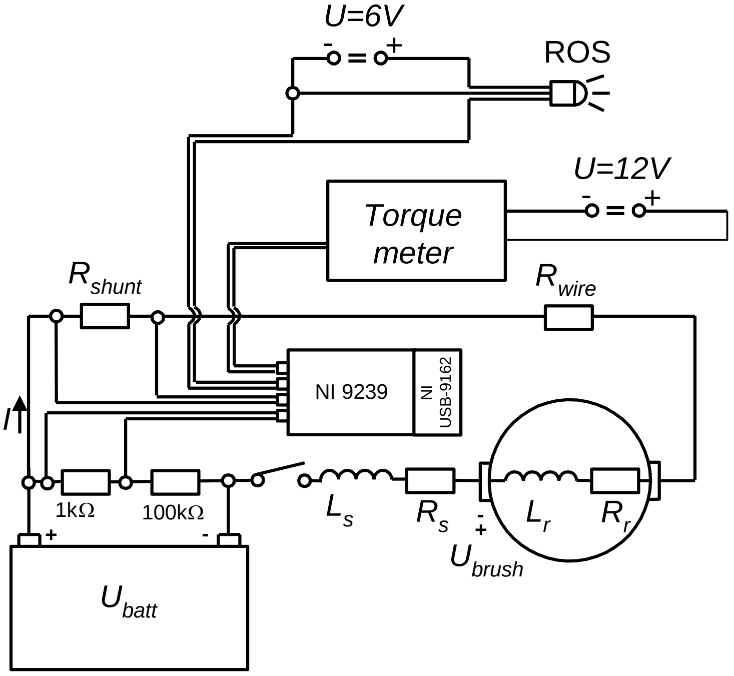


Figure 2. Wiring diagram of the Hardware Component of the Measurement System

The angular speed of the rotor is measured by an optical led sensor – *Model: ROS-P, Monarch Instrument*. The laser light emitted by the instrument is reflected by small foils fixed on the circumference of the rotor at equal distances. The reflected light then hits the sensor inside the instrument which generates a voltage signal in its output. The height of the output voltage signal is equal to the supply voltage of the sensor ($U_{OLS} = 6\ V$). A typical output voltage signal of the optical led sensor vs. time can be seen in Figure 3.

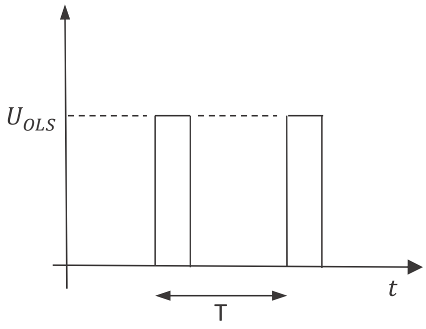


Figure 3. An output voltage signal of the optical led sensor vs. time

The time gap between two neighboring signals is calculated as:

$$T = \frac{60}{n \cdot N} \quad (1)$$

Where n is the RPM of the motor, and N is the number of the light reflecting foils fixed on the rotor. The torque of the motor is measured by a rotary torque transducer – *HBM, T22/200 Nm* –, with an accuracy class of 0.5. The applied torque on this transducer is linearly proportional to its output voltage (U_{RTD}):

$$M = C \cdot U_{RTD} \ (2)$$

The value of constant C is equal to $40\ \mathrm{Nm/V}$. The intensity of the electric current flowing through the windings of the motor is measured as a voltage drop on a shunt resistor with an electric resistance (R_{SHUNT}). Thus, the intensity of the electric current is calculated from the measured output voltage as:

$$I = \frac{U_{SHUNT}}{R_{SHUNT}} \ (3)$$

The terminal voltage of the battery – since it is usually bigger than the maximum allowed input voltage of the *NI 9239* module –10 V – is measured by applying a voltage divider with a ratio of M : N, typically 1:100. The connection between the output voltage measured on the divider (U_{DIV}) and the terminal voltage of the battery is:

$$U_{BATTERY} = \frac{M+N}{M} \cdot U_{DIV} \ (4)$$

3. Description of the Software Component

As it is explained above, the MS consists of two parts the HC and the SC. The main purposes of the SC are to separate, collect, store, process, analyze and graphically display the measured and the calculated data. In Figure 4. the structure of the SC is presented.

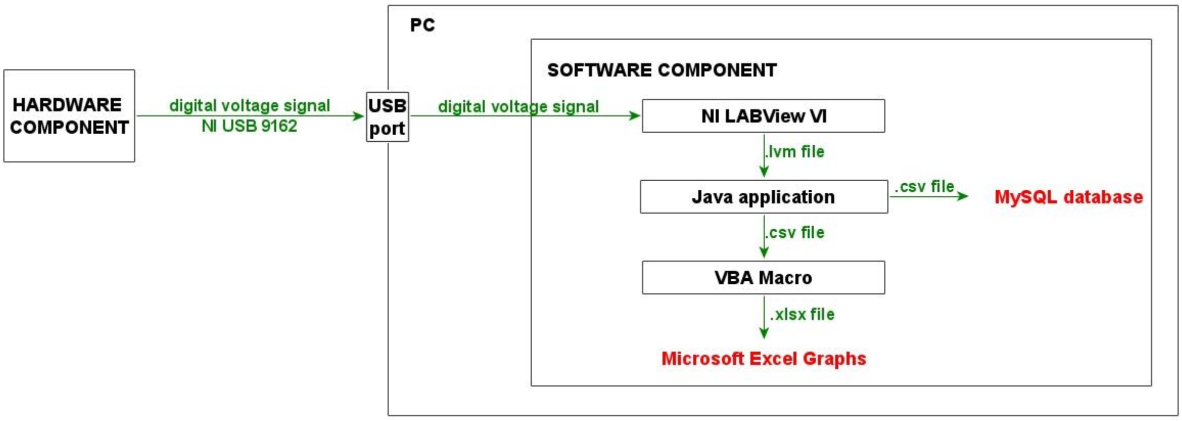


Figure 4. Structure of the Software Component

The SC contains four different modules; the first one was implemented in NI LabVIEW and provides the collection, separation, and text file output of the measured data that is coming from the sensors. Furthermore, it is capable of displaying both the raw voltage signals graphically, and the calculated output values numerically in real-time. The Front Panel of the VI serves as a User Interface for the whole SC. The second module is a command-line application that was implemented in Java SE [9], with the help of the Apache NetBeans IDE. The Java application provides the tool to pre-process aka clean and filter the raw data collected by the NI LabVIEW VI, and to make the connection – insert and retrieve data – with the underlying MySQL database. The third module is a VBA Macro, which was created for Microsoft Excel, which is the most frequently used software by engineers, although it is not commonly referred to as an engineering software by definition. Excel itself has limitations when it comes to processing a huge amount of data. That is why it was necessary to insert the command-line Java application into the process flow. For example, it took more than ten minutes (that was the upper limit of the developer's patience) for Microsoft Excel to filter an NI LabVIEW measurement file – which consists of almost three hundred thousand lines; while the Java command-line application solved the same problem in 1 second. The VBA Macro generates the desired output values and visualizes them, by using Microsoft Excel charts. The fourth module is the MySQL database, which is a simple OLTP database, with two tables to provide a long-term storage place for the measured data.

3.1. NI LabVIEW VI

Four analog input channels are used for collecting analog voltage signals by the NI 9239 measurement card. The channels collect signals from different sensors which are then converted to a digital signal by the built-in analog-digital converter of the card. The input circuitry of the measurement card can be seen in Figure 5.

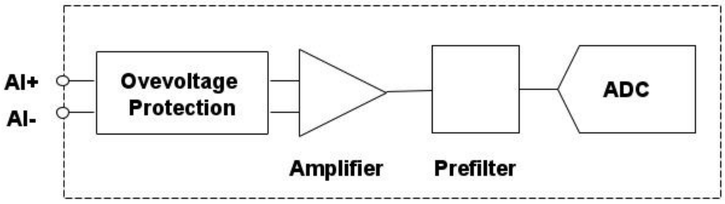


Figure 5. The input circuitry of the NI 9239

The resulting digital voltage signal serves as an input for the NI LabVIEW VI. To be able to process the input, first it is necessary to insert a DAQ I\O control for the physical channel in the Block Diagram of the VI. Several solutions are available in the NI LabVIEW to retrieve data from a physical channel, and in the NI DAQmx collection as well. For faster processing, the step-by-step solution was chosen by the developer instead of the DAQAssistant; which provides an off-the-shelf, but much slower solution. The step-by-step solution gives more freedom when it comes to deciding how to process the input data. Typical steps of data processing with the DAQmx VI-s can be seen in Figure 6. The displayed data flow reads from a physical channel which transmits a voltage signal on a single channel. The processing steps are the following according to Figure 6.:

- Firstly, a virtual channel is made for acquiring voltage input from the physical channel.
- Secondly, the properties of the reading process are set by a timing VI. The sampling rate and mode must be given here.
- Thirdly, – before reading – a data acquisition task is started.
- Fourthly, the suitable read VI – number of channels and number of samples fit to the physical channel – is called to read data with different purposes.
- Fifthly, methods and operations are executed to meet the purpose of the VI. In that particular example, the measured data is written into a(n) NI LabVIEW measurement file.
- Sixthly, when the reading is finished the task has to be stopped and cleared.

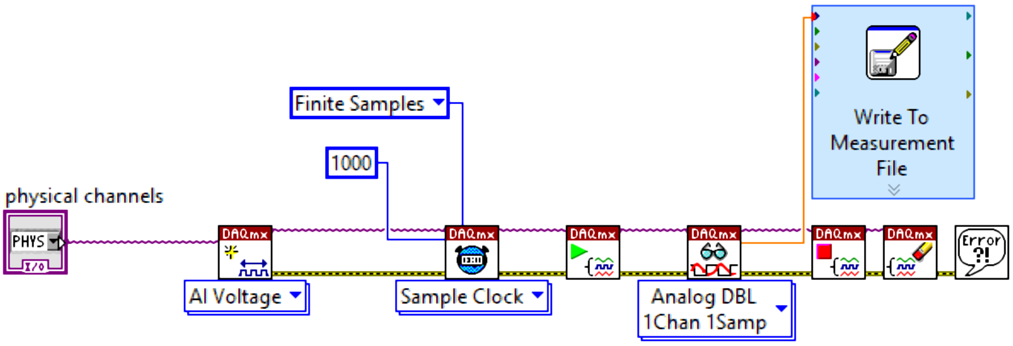
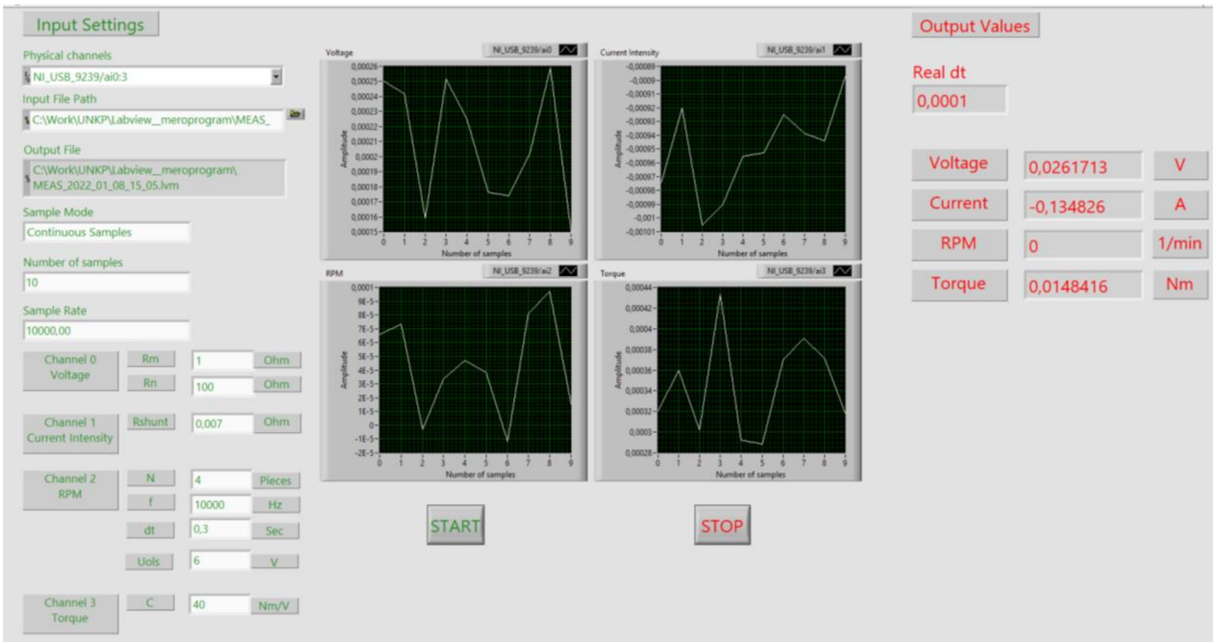


Figure 6. Typical data process flow with the DAQmx VI collection (voltage signal, single-channel, finite sampling mode and 1000 sample rate)

In our case, not one, but four voltage signals have to be processed because the measurement card has four analog inputs. Thus, instead of using a one-channel one-sample type of read VI, a multi-channel multi-sample type was used. The NI LabVIEW VI is capable of separating the data arriving from the four different output channels and displaying them on waveform charts versus the number of samples in the actual loop. Also, by applying some data manipulation methods and mathematical operations – with predefined input parameters, set on the User Interface, – the voltage, current, RPM, and torque output values can be calculated and displayed in indicators on the Front Panel of the self-developed VI. The VI provides collect and store functionality, by writing the measured data into the NI LabVIEW measurement file.



Input Settings

Physical channels: NI_USB_9239/ai0:3

Input File Path: C:\Work\UNKP\Labview_meroprogram\MEAS_

Output File: C:\Work\UNKP\Labview_meroprogram\MEAS_2022_01_08_15_05.lvm

Sample Mode: Continuous Samples

Number of samples: 10

Sample Rate: 10000,00

Channel 0 Voltage	Rm	1	Ohm
	Rn	100	Ohm
Channel 1 Current Intensity	Rshunt	0,007	Ohm
Channel 2 RPM	N	4	Pieces
	f	10000	Hz
	dt	0,3	Sec
	Uols	6	V
Channel 3 Torque	C	40	Nm/V

Output Values

Real dt: 0,0001

Voltage	0,0261713	V
Current	-0,134826	A
RPM	0	1/min
Torque	0,0148416	Nm

Figure 7. User Interface aka Front Panel of the self-developed NI LabVIEW VI

3.2. *Java command-line application*

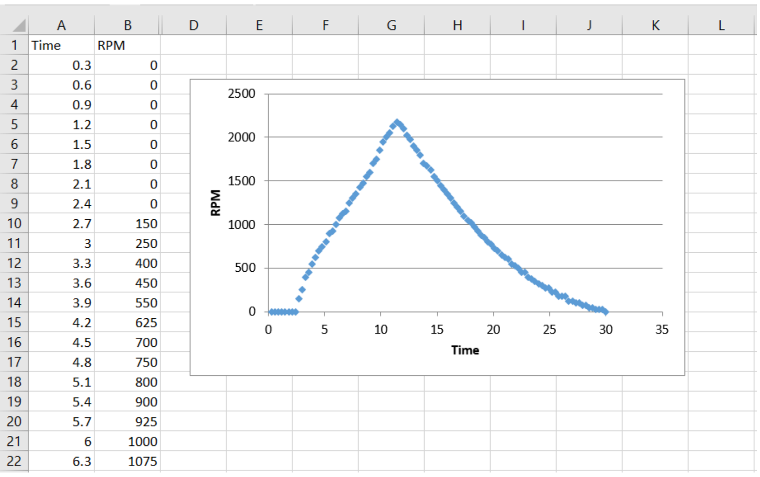
The output data of the above-mentioned NI LabVIEW VI is one or more NI LabVIEW measurement files, which – according to the length of the measurement – can contain a few, but mostly a huge quantity of measured data in rows. Every row contains a time column and one to the four measured value columns, based on the number of input channels used. The developed Java command-line application inputs are the aforesaid NI LabVIEW measurement files (.lvm). Two main tasks have to be realized by the java application:

- The first is to insert measured values from all of the row input found in the NI LabVIEW measurement file/files into the underlying MySQL database, thus realizing the store feature of the SC.
- The second is to filter the rows based on the time gap between two measurements (dt); given by the user on the User Interface of the SC, to pre-process the data.

In default, measurements are recorded every 0.0001 seconds, by the self-developed VI. By applying a filter in the time gap values, a much smaller file size can be achieved. The third step of the process method is analyzing the data with Microsoft Excel. As it is mentioned earlier above, the application cannot deal easily with too many lines. The Java command-line application converts the input NI LabVIEW measurement files into 'comma-delimited' text files (.csv) because of compatibility issues.

3.3. VBA Macro

Microsoft Excel is easily automatable with the help of macros written in Visual Basic for Applications. It is a high-level, object-oriented programming language. The main task of the developed VBA Macro is to **analyze and display the processed results graphically** in charts. When the "comma-separated values" text file is opened in Microsoft Excel, the macro converts it into a Microsoft Excel workbook (.xlsx). Then, based on the input parameters given in the User Interface of the SC, all of the calculations – explained previously in Section 2 – are done, to gain the desired output values, like the voltage, current intensity, RPM, and/or the torque. As the last step, one or more graphs are built and displayed to be viewed later for investigation of the output values from the experimental tests. In Figure 8. one output Microsoft Excel workbook is presented, which was made during an experiment where the RPM needed to be calculated.

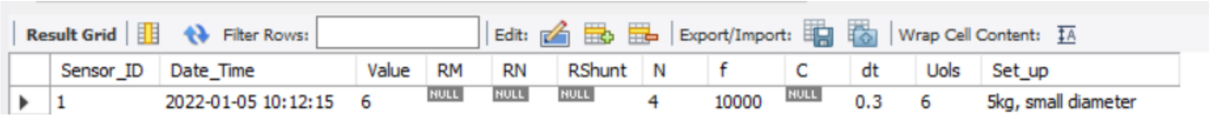


Time	RPM
0.3	0
0.6	0
0.9	0
1.2	0
1.5	0
1.8	0
2.1	0
2.4	0
2.7	150
3	250
3.3	400
3.6	450
3.9	550
4.2	625
4.5	700
4.8	750
5.1	800
5.4	900
5.7	925
6	1000
6.3	1075

Figure 8. Results of test measurement for RPM with the self-developed VBA Macro

3.4. MySQL database

The MySQL database serves as a long-term storage place for the measured data, collected during different experimental test measurements. The collected data can provide a solid base for statistical research about the investigated motors, or the properties and behavior of the applied sensors as well. It is a relational database that contains two tables connected by a foreign key – which is labeled as sensor_id and id in the two tables, they share the same data type (which is a string with a length of thirty). The foreign key is important because it restricts the storage of any measured data into the database, if the corresponding sensor type is nonexistent in the sensor table. The model of the database is shown in Figure 10. First of all, it was necessary to design a table to store all the vital data about the different sensors available in the MS, like the type, manufacturer, measured unit, or the valid operating range of it. **Storing** information like the previously mentioned parameters is useful because several errors can be found without any human interactions during measurements. The main table of the database contains all of the previously measured data during experimental tests. The records are identified by the sensor_id which was used to measure the value stored in the actual record, along with the date- time attribute, which consists of the date and the precise time when the measurement was taken. Every row contains only one measured value and the corresponding input parameters. The others remain empty. For example, 6V is measured during an RPM test, then the record will be the following:



Sensor_ID	Date_Time	Value	RM	RN	RShunt	N	f	C	dt	Uols	Set_up
1	2022-01-05 10:12:15	6	NULL	NULL	NULL	4	10000	NULL	0.3	6	5kg, small diameter

Figure 9. Sample record of the Measured Data table

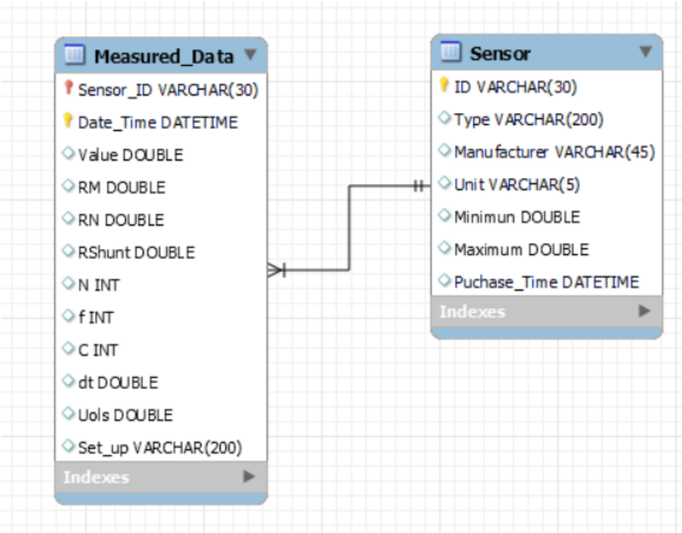


Figure 10. Database model of the underlying MySQL database of the Software Component

4. Summary

The Software Component of an electric motor test bench was realized in the NI LabVIEW environment. The main purposes of the developed Software Component are to separate, collect, store, process, analyze and graphically display the data measured from an electric motor. These data are the angular speed, torque, voltage of the motor, and the intensity of the electric current flowing through it. The NI LabVIEW VI was completed with a Java command-line application for faster pre-processing, a VBA Macro for data analysis, and a MySQL database which serves as a long-term storage place for the measured data. The developed Software Component was partly tested during measurements on a three-phase induction motor, and it has met the requirements while proving to be user-friendly. After having more experience with the Measurement System, we intend to realize additional improvements on the Software Component and complete it with additional features answering the arising demands during the experimental work.

Acknowledgment

„SUPPORTED BY THE ÚNKP-21-3 NEW NATIONAL EXCELLENCE PROGRAM OF THE MINISTRY FOR INNOVATION AND TECHNOLOGY FROM THE SOURCE OF THE NATIONAL RESEARCH, DEVELOPMENT AND INNOVATION FUND.”

„SUPPORTED BY THE ÚNKP-21-4-II NEW NATIONAL EXCELLENCE PROGRAM OF THE MINISTRY FOR INNOVATION AND TECHNOLOGY FROM THE SOURCE OF THE NATIONAL RESEARCH, DEVELOPMENT AND INNOVATION FUND.”

"THE RESEARCH WAS SUPPORTED BY THE THEMATIC EXCELLENCE PROGRAMME (TKP2020-NKA-04) OF THE MINISTRY FOR INNOVATION AND TECHNOLOGY IN HUNGARY."

References

[1] Szántó, Attila, Sándor Hajdu, and Gusztáv Áron Sziki. "Dynamic simulation of a prototype race car driven by series wound DC motor in Matlab-Simulink." Acta Polytech. Hung 17.4 (2020): 103-22.

[2] Szíki, Gusztáv Áron, Attila Szántó, and Tamás Mankovits. "Dynamic modelling and simulation of a prototype race car in MATLAB/Simulink applying different types of electric motors." International Review of Applied Sciences and Engineering 12.1 (2021): 57-63.

[3] Hadžiselimović, Miralem, et al. "Magnetically nonlinear dynamic model of a series wound DC motor." Przegląd Elektrotechniczny 87.12b (2011): 60-64.

[4] Szántó, Attila, András Szántó, and Gusztáv Áron Sziki. "Review of the modelling methods of series wound DC motors." Műszaki Tudományos Közlemények 13.1 (2020): 166-169.

[5] Sziki, Gusztáv Áron, et al. "Experimental investigation of a series wound dc motor for modeling purpose in electric vehicles and mechatronics systems." Measurement 109 (2017): 111-118.

[6] Szántó, A. et.al: Dynamic Test Measurements and Simulation on a SeriesWound DC Motor. Appl. Sci. 2021, 11, 4542. https://doi.org/10.3390/app11104542

[7] What is LabVIEW?_ https://www.ni.com/hu-hu/shop/labview.html

[8] Elliott, C., Vijayakumar, V., Zink, W., & Hansen, R. (2007). National instruments LabVIEW: a programming environment for laboratory automation and measurement. JALA: Journal of the Association for Laboratory Automation, 12(1), 17-24.

[9] Schildt, H., & Coward, D. (2014). Java: the complete reference (p. 1312). New York: McGraw-Hill Education.