Data Acquisition and Control System Based on Scilab Software Environment

Marek Godlewski, Krzysztof Rogowski
Politechnika Białostocka, Wydział Elektryczny, ul. Wiejska 45D, 15-351 Białystok

Abstract: The article presents the data acquisition and control system built using the Scilab software environment which can be an interesting alternative for commercial and very expensive solutions. The system is composed of a National Instruments myDAQ data acquisition device connected to a PC with open-source software for numerical computations Scilab. The proposed system is equipped with Graphical User Interface (GUI) that allows users to process control, visualize and save the acquired data. In addition, the PID control strategy is implemented. Finally, results of laboratory tests verifying the operation of the created system and conclusions are presented.

Keywords: data acquisition, Scilab, measurement, PID control

1. Introduction

Nowadays, Data Acquisition systems (DAQ) enabling the collection of measurement data of dynamic objects, allowing to monitor and control of the operation of a given object are increasingly used in various fields of life, especially in industry and science. This is due to the need to record more and more data to make developmental business decisions. It is therefore a desirable issue in every well-developing industrial company, as it allows one to avoid the occurrence of problems, time losses, faster repair of damages, or more intensive development. Therefore, for such a system to meet the requirements and demands of users, it must be characterized by reliability, accuracy, and ease of use [2, 3, 7, 8].

There are many DAQ systems available on the market, however, most of them require possessing very expensive software like LabVIEW, MATLAB, etc. Therefore, for less demanding measurement problems there is a need to find some cheaper alternatives for commercial solutions.

The Scilab environment is an accessible open-source software for performing numerical calculations, having many methods tailored for this purpose, including solving linear systems, determining eigenvalues/vectors, fast Fourier transform, solving differential equations, optimization algorithms, etc. Additionally, Scilab has functions for creating low- and high-level graphics. The programming language introduced in this environment that operates on matrices is a simple, powerful, and efficient tool. This software can also work with subroutines written in C, C++ or Fortran. Except for a few commands related to graphics, the syntax of Scilab is the same as that used in MATLAB [6, 9, 10].

In [4, 5], a similar system was built using RTAI-Lab environment and Scilab. The presented tool requires the Linux kernel to be installed with ARM-based device and is dedicated to rapid control system of a DC servo motor.

In this paper, the Scilab environment will be used to develop the author's data acquisition system. The system will be based on the popular NI myDAQ measurement card and toolboxes: NIDAQ and GUI Builder. It will consist of two programs: the first one, enabling only data acquisition and visualization, and the second one, which will be extended with the implementation of a PID controller.

2. Scilab and required toolboxes

Scilab was developed in 1990 by the French National Computer Research Institute INRIA and the world's oldest engineering school ENPC for engineers and scientists. It has been distributed since 1994 under an open-source license [9, 10]. It is also licensed under the GPL, which allows the user much freedom, such as using the software for any purpose and adapting it to his needs or sharing the software with other users.

Scilab is primarily used for mathematical computations and analysis. It allows operations on numbers as well as on vectors or matrices and contains hundreds of mathematical functions. Thanks to the presence of a high-level programming language in Scilab, it is possible to access advanced data structures or graphical functions. It can also cooperate with programs written in other languages. It can be used in many fields, from simple calculations to signal processing or process control [9, 10].

The NIDAQ toolbox enables Scilab to work with National Instruments devices. This module allows the user to [10]:
- easily configure the measuring device and read the data in Scilab for immediate analysis,
- send data through analog and digital output channels of a measuring device,

Autor korespondujący:
Krzysztof Rogowski, k.rogowski@pb.edu.pl

Artykuł recenzowany
nadesłany 30.06.2022 r., przyjęty do druku 27.09.2022 r.

– control the analog input and output, counter/timer and digital I/O of a DAQ device (National Instruments).

NIDAQ module offers the functions such as [10]:
– configuration and creation of channels for the measurement card – *DAQ_CreateAIVoltageChan* for analog inputs and analog outputs *DAQ_CreateAOVoltageChan*,
– input data to the system from the HMI - *DAQ_WriteAnalogF64* command,
– reading measured data into the system using the *DAQ_ReadAnalogF64* command,
– configuration and control of tasks in the system – *DAQ_CreateTask*, *DAQ_StartTask*, *DAQ_StopTask* and *DAQ_ClearTask* commands,
– errors handling – DAQ_ErrChk command, which checks the status of the operation, and in case of failure the associated task specified by *taskHandle* is stopped, deleted, and the corresponding error message appears
– getting some information about the system with the *DAQ_SystemInfoAttribute* command.

The latest version of the NIDAQ module (version 0.3.2) was released in 2012 and is compatible with the Scilab 5.4.x version (for Windows 32- or 64bit).

GUI Builder is a module used to build graphical user interfaces for applications created in Scilab [12]. The program allows the user to quickly build a GUI by adding objects, resizing them and moving them around in the window. It also allows for modification of their properties using the Object Properties button or in the code, which is automatically generated for the built interface. The user building the graphical interface with GUI Builder has at his disposal such objects as [12]:
– pushbutton – a rectangular button usually used to trigger callbacks,
– radiobutton – a button with two states,
– checkbox – button with two states (used for many independent selections),
– edit – editable string area,
– text – text control,
– slider – a scale control, which is a slider used for setting values between ranges with the use of mouse,
– frame – a space for inserting other objects,
– listbox – a control representing a list of items that can be scrolled and selected with the mouse,
– popupmenu – a button that causes the display of a menu when it is clicked.

A view of the GUI Builder windows for Scilab version 5.4.1 is shown in Figure 1.

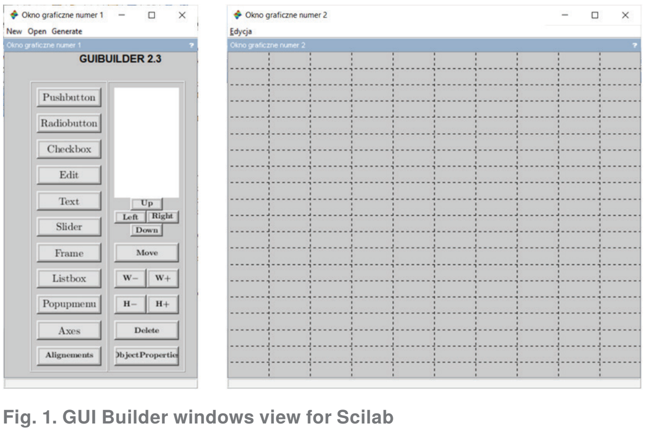

Fig. 1. GUI Builder windows view for Scilab
Rys 1. Widok okien GUI Builder dla programu Scilab

3. NI myDAQ measuring card

NI myDAQ is a portable data acquisition device designed and produced by National Instruments that allows the user to take measurements and analyze them in the real world. By combining NI myDAQ with the software on a computer, it is possible to analyze and process the signals obtained and to control simple processes anywhere and anytime [11]. An illustration of the NI myDAQ measurement card is presented in Figure 2.

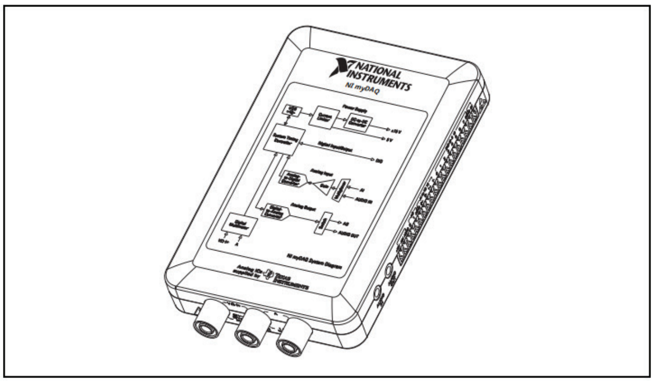

Fig. 2. NI myDAQ measurement card [13]
Rys. 2. Karta pomiarowa NI myDAQ [13]

NI myDAQ has two analog inputs AI, two analog outputs AO, eight-bit digital input and output DIO, two audio inputs, three power inputs, digital multimeter input, and a USB connector to connect it to a computer [13]. Therefore, it offers many possibilities in the area of measurement and control.

In the proposed DAQ system described in this paper, one analog input and one analog output will be used. The most important parameters of the AI analog inputs are [13]:
– ADC resolution – 16 bits,
– maximum sampling frequency – 200 kS/s,
– accuracy of time measurement – 100 ppm of the sampling frequency,
– voltage range – ±10 V.

Key parameters of the AO analog outputs of the NI myDAQ measurement card [13]:
– DAC resolution – 16 bits,
– maximum update rate – 200 kS/s,
– voltage range – ±10 V.

4. Implementation of DAQ in Scilab

The first stage in the execution of the algorithm was to create a graphical interface using the previously installed GUI Builder toolbox and then generate code for it to add appropriate action functions to each of the objects used.

For the program used solely for data acquisition and visualization, the graphical interface was equipped with three pushbutton objects, two text objects and one slider and axes object each. The completed graphical window is shown in Figure 3.

Using the START button, the user initiates the start of the measurement process. In the code of this object, there are initialized parameters and determination of analog input and output channels and ranges of the measurement card, which will be used for the measurements. This is followed by a piece of code responsible for performing the measurements, assigning the collected measurement points to an array, and visualizing them on a graph in the axes object. The whole process of performing measurements depends on the STOP variable associated with

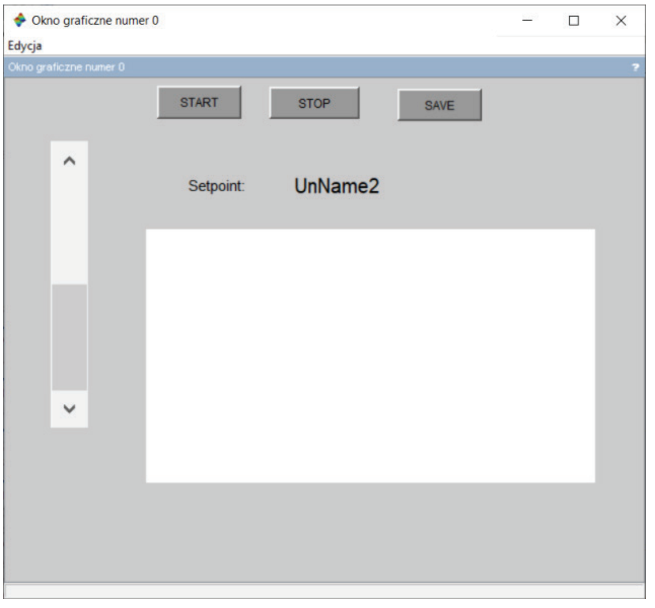

Fig. 3. The graphical window of the system performing data acquisition and visualization
Rys. 3. Okno graficzne systemu wykonującego akwizycję i wizualizację danych

the button of the same name, it must have a logical value equal to 0 – false. Subsequent measurements are made using a while loop. The values of the measurement points (red) and the set value (blue) for which the measurements are obtained. Pressing the STOP button causes the logical value of the STOP variable to change, which in turn causes the measurements to stop. The last button – SAVE – enables saving the points from performed measurements as well as the setpoint and time-instant values for which the measurements were performed to a .txt text file. They are presented in the form of three columns. The first column shows the values of the measurements performed (AI values). The second column is a vector created with the same length as the first one and contains the values set on the DAQ output (AO). The third column contains time-instant values in seconds. The created file can be given any name and saved anywhere on the disk. The slider is used to put the voltage on the analog output of the measurement card. The range of changes of this voltage was declared in the properties during its addition and is from 0 to 10V. The value of the voltage set from the slider is assigned to the variable *data_ao* and displayed in one of the added text objects.

5. PID and program control options

The next functionality of the proposed DAQ system is the possibility of PID control of dynamical models. For this purpose, the discrete-time PID controller equations were implemented in the system.

In the code of the system algorithm global variables describing the PID controller parameters were added, and their initial values assigned ($K_p = 1$, $K_i = 1$, $K_d = 0$). The functioning of some of the used graphic objects (START button, SAVE button) has also been changed, and appropriate functions for inserting new values into edit fields have been implemented.

In the function referring to the START pushbutton all global variables used in the PID calculations responsible for ensuring zero initial conditions and the range of output voltages of the regulator were introduced. The calculated coefficients of the input and output signal equations were there implemented.

The PID control strategy is implemented as a discrete-time version of the dynamic equation of the PID controller [1]

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t), \tag{1}$$

where K_p is a gain of proportional part and K_i, K_d are gains of integral and derivative part, respectively.

The transfer function representation of the PID controller is given by

$$C(s) = K_p + \frac{K_i}{s} + \frac{NK_d}{1 + N/s}, \tag{2}$$

where N is a constant of a low-pass filter.

By the discretisation of integral I and differential D using the backward Euler method we obtain the discrete-time form of the PID equation

$$C(z) = \frac{U(z)}{E(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}}, \tag{3}$$

where the coefficients b_0, b_1, b_2, a_0, a_1, a_2 are described by the following equations

$$\begin{aligned}
b_0 &= K_p(1 + NT_s) + K_i T_s (1 + NT_s) + K_d N, \\
b_1 &= -\left(K_p(2 + NT_s) + K_i T_s + 2K_d N\right), \\
b_2 &= K_p + K_d N, \\
a_0 &= (1 + NT_s), \\
a_1 &= -(2 + NT_s), \\
a_2 &= 1.
\end{aligned} \tag{4}$$

Finally, the coefficients of the controller output signal were calculated from the transformations of the PID differential equation:

$$a_0 U(z) + a_1 z^{-1} U(z) + a_2 z^{-2} U(z) = b_0 E(z) + b_1 z^{-1} E(z) + b_2 z^{-2} E(z).$$

Hence, we get the recursive equation of the PID control

$$u[k] = -\frac{a_1}{a_0} u[k-1] - \frac{a_2}{a_0} u[k-2] + \frac{b_0}{a_0} e[k] + \frac{b_1}{a_0} e[k-1] + \frac{b_2}{a_0} e[k-2].$$

In the algorithm, a vector of $u[k]$ was also created to collect all the history of the past control samples. The graph shows three signals: the first set comes from the slider, and this is a set point, the control signal from the controller output, and the signal measured on the output of the plant (the controlled signal).

In the further part of the algorithm assigned to the START button – a controller feedback loop was implemented in the while loop. This was done using the controller's output signal equation and by assigning newly calculated input and output signal values to their previous values. Care was also taken to ensure that the signals do not fall outside the declared voltage range (from 0 V to 10 V) by conditionally assigning maximum or minimum values to those values (outside the range). This allowed for the fully correct operation of the built PID controller.

The functioning of the SAVE button has also been changed very slightly. In the structure of the whole code, the write function has been replaced by the *fprintfMat* function (the write function only allows three columns of data to be written). The change was caused by extending the saved file with the recorded control values of the PID controller. The control signal points were placed in the third column of the .txt file, while the time vector points went to the fourth column.

Compared to the code of the program used for data acquisition and visualization, the implementation of the functionality of graphic edit objects used to change the controller settings (K_p, K_i, K_d) was completely new. In the first step, the declared global variables K_p, K_i, and K_d were handled by the corresponding edit fields. However, to use the inserted values of the fields correctly throughout the algorithm, they had to be converted from the string variable to double, which was done using the *strtod* function.

6. Results of laboratory tests

The practical verification of the developed systems for data acquisition and visualization and PID control will consist, respectively, of the registration of the step characteristic of a dynamic plant and two control characteristics. As the plant the laboratory model show in Figure 4 in taken.

Fig. 4. Front panel of the dynamic system model
Rys. 4. Płyta czołowa modelu obiektu dynamicznego

The set value for both registered characteristics is $U_o = 4.623$ V. For one of the control characteristics, the values of the PID controller set points remain intact in relation to the initial ones. The second one will be performed for the values of the settings determined by the Ziegler-Nichols graphical method based on the plant step characteristic. As a dynamic object used to verify the operation of the programs, the third-order inertial plant was chosen described by the transfer function

$$G(s) = \frac{1.89}{(5s+1)(s+1)(2.5s+1)}.$$

6.1. Step characteristics of a dynamic object and writing data to a text file

Figure 5 shows the waveforms of two signals: the signal of set voltage (blue color) and the step characteristic consisting of measurement points (red color). The step characteristic in its shape fully reflects the character of the inertial plant of the third order.

Saving the data to a text file was done by pressing the SAVE button during the recording of the step characteristics to check the correctness of the data acquisition and visualization software. From the saved data, an identical characteristic is obtained as in Figure 5.

6.2. PID control characteristic

In the next step, the same three-inertia dynamical system will be controlled by the implemented in the DAQ system PID controller.

Figure 6 shows three signals: the set-point signal (blue), the control signal (green), and the output - measurement signal, which forms the control characteristic (red). The obtained control characteristic recorded for the same setpoint and transmittance of the inertial dynamical system of the third-order allows one to assess the quality of regulation. As can be seen, the application of the PID regulator with the default settings values ($K_p = 1$, $K_i = 1$, $K_d = 0$) made it possible to obtain the regulation of the measurement signal regulation: zero steady-state

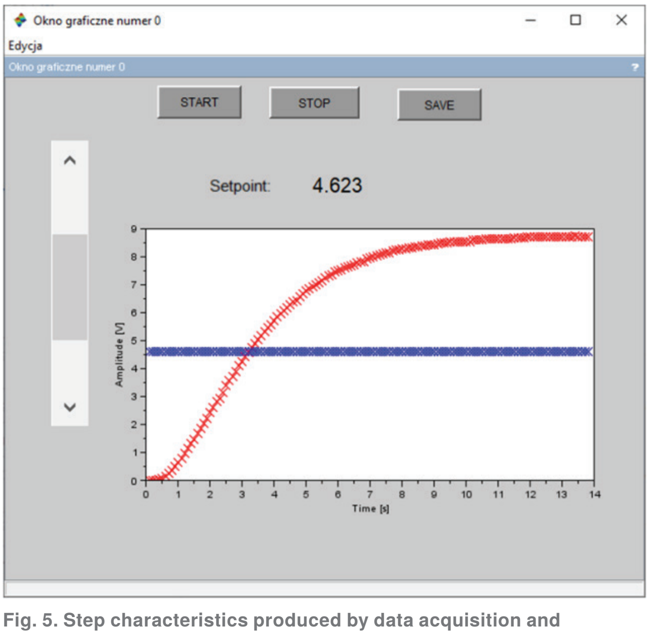

Fig. 5. Step characteristics produced by data acquisition and visualization software
Rys. 5. Charakterystyka skokowa wykonana przy użyciu programu służącego do akwizycji i wizualizacji danych

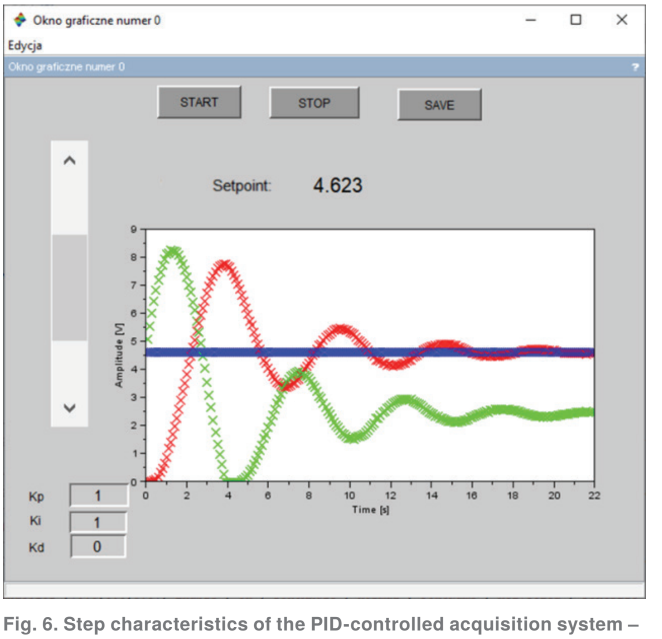

Fig. 6. Step characteristics of the PID-controlled acquisition system – default settings of the PID controller
Rys. 6. Charakterystyka regulacyjna systemu akwizycji z regulacją PID – domyślne nastawy regulatora PID

error determined by the integration part of the PID controller, overshoot of about 68 % and settling time of 15.1 seconds.

These results will be compared with the PID controller with parameters tuned using Ziegler-Nichols open-loop method [1]. First, it was necessary to determine the parameters R and L based on the step characteristics shown in Figure 5.

The parameters values of the controlled are computed as follows

$$K_p = \frac{1.2}{0.75 \cdot 1.94} \approx 0.69$$

$$T_i = 2 \cdot 0.75 = 1.5$$

$$K_i = \frac{K_p}{T_i} = \frac{0.69}{1.5} \approx 0.55$$

$$T_d = 0.5 \cdot 0.75 = 0.375$$

$$K_d = K_p \cdot T_d = 0.69 \cdot 0.375 \approx 0.31$$

The step characteristic of a control system with parameters of the PID controller tuned by the open-loop Ziegler-Nichols are shown in Figure 7.

The control characteristic shown in Figure 7, shows a better quality of plant regulation compared to the characteristic shown in Figure 6. A more detailed assessment of the quality of the performed regulation confirms the observations, i.e., zero steady-state error of control, overshoot of about 39 %, and settling time of 8.6 seconds.

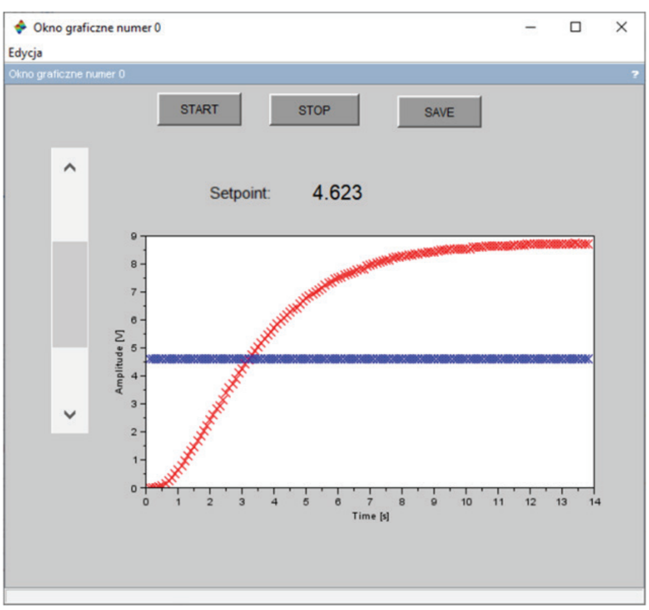

Fig. 7. Control characteristics of the acquisition system with PID control – PID controller set values selected using the open-loop Ziegler-Nichols method

Rys. 7. Charakterystyka regulacyjna systemu akwizycji z regulacją PID – wartości nastaw regulatora PID dobrane metodą charakterystyki skokowej wg Zieglera-Nicholsa

7. Conclusions

The constructed acquisition system allows one to measure the data from one object. The user, using a graphic window, can set and change the setpoint value of the signal, observe the obtained measurement results in real-time, and save the data to a text file for in-depth analysis performed in another program. It can also change the PID regulator settings values to obtain the most accurate regulation of the output signal. The construction of the data acquisition system using the myDAQ NI measurement card forced the use of an older version of Scilab software – Scilab 5.4.1. The reason for this choice was the lack of availability of an additional module in the latest version that would enable communication between the card and the software environment.

Experimental verification of the constructed system shows the accuracy of the programs executed in the Scilab environment. The recorded step characteristics fully reflect the character of the dynamic object – a 3rd order inertial segment. The characteristic carried out in MATLAB based on data stored in a text file coincides with the registered one. Performing the characteristic curve on this dynamic object, which is the system of automatic regulation, the inertial segment of the third order with the same transmittance and the same value of the set voltage, allows one to notice the regulation of the measurement signal. However, the selection of PID controller settings by the step characteristic method according to Ziegler-Nichols enables us to obtain better indicators of the performed regulation.

Acknowledgements

This research was carried out in the framework of the grant no. WZ/WE-IA/6/2020 financed from the funds for science by the Polish Ministry of Science and Higher Education.

References

1. Duzinkiewicz K., Grochowski M., Piotrowski R., *Regulatory PID*, Politechnika Gdańska Wydział Elektrotechniki i Automatyki, 2009.
2. Emilio M.P., *Data Acquisition Systems: From Fundamentals to Applied Design*, Springer, 2013, DOI: 10.1007/978-1-4614-4214-1.
3. Park J., Mackay S., *Practical Data Acquisition for Instrumentation and Control Systems*, Elsevier, 2003, DOI: 10.1016/B978-0-7506-5796-9.X5000-9.
4. Skiba G., Żabiński T., Wiktorowicz K., *Rapid prototyping of servo controllers int RTAI-Lab*, VI Conference on Computer Methods and System, Cracow 21-23 November 2007, 141–146.
5. Skiba G., Żabiński T., Bożek A., *Rapid Control Prototyping with Scilab/Scicos/RTAI for PC-based and ARM-based Platforms*, 2008 International Multiconference on Computer Science and Information Technology, DOI: 10.1109/IMCSIT.2008.4747324.
6. Urroz G.E, *Introduction to SCILAB*, InfoClearinghouse.com, 2001.

Other sources

7. National Instruments, *Data Acquisition Fundamentals*, Application Note 007, National Instruments Corporation, 2002.
8. https://mikrokontroler.pl/2016/03/08/nowoczesne-systemy-akwizycji-danych/ – Data acquisition systems for beginners,
9. https://pl.wikipedia.org/wiki/Scilab – Information about Scilab
10. https://www.scilab.org/about/scilab-open-source-software, Scilab as open-source,
11. https://atoms.scilab.org/toolboxes/Nidaq/0.3.2 – NIDAQ module description.
12. https://atoms.scilab.org/toolboxes/guibuilder/4.2.1 – GUI Builder module description.
13. https://www.ni.com/docs/en-US/bundle/mydaq-features/resource/373060g.pdf – Datasheet for the myDAQ NI measurement card.

System akwizycji danych i sterowania z wykorzystaniem środowiska programowego Scilab

Streszczenie: W artykule zaprezentowano system akwizycji danych i sterowania zbudowany w oparciu o środowisko programowe Scilab, który może stanowić interesującą alternatywę dla komercyjnych i bardzo kosztownych rozwiązań dostępnych na rynku. Układ składa się z karty akwizycji danych myDAQ produkowanej przez National Instruments oraz komputera wyposażonego w oprogramowanie do obliczeń numerycznych Scilab. Stworzono graficzny interfejs użytkownika (GUI), który pozwala na kontrolę procesu, wizualizację i zapis zmierzonych danych. Dodatkowo zrealizowano układ regulacji automatycznej z regulatorem PID. Na koniec pokazano badania laboratoryjne weryfikujące poprawność działania systemu oraz przedstawiono wnioski.

Słowa kluczowe: akwizycja danych, Scilab, pomiary, regulacja PID

Marek Godlewski, Eng.
marekgodlewski09@gmail.com
ORCID: 0000-0002-8031-8283

In 2022, he received an engineering degree in electronics and telecommunications at the Faculty of Electrical Engineering of Bialystok University of Technology – specialization in industrial electronics and electronic apparatus. He is currently studying automation and robotics at the Faculty of Mechanical Engineering of Bialystok University of Technology and is working on issues related to this field.

Krzysztof Rogowski, PhD, Eng
k.rogowski@pb.edu.pl
ORCID: 0000-0002-6871-4806

He received the MSc (2007) and PhD (2012) degrees from Bialystok University of Technology. He is currently an Adjunct Professor with Faculty of Electrical Engineering, Bialystok University of Technology. His current research interests include modern control theory, multidimensional systems, and fractional order systems.