VIRTUAL LABORATORY IN THE ENGINEERING EDUCATION

Maciej Drabik[1] PhD, **Alina Gil**[2] PhD, **Urszula Nowacka**[3] PhD
Jan Dlugosz University, Czestochowa, Poland[1,2,3];
m.drabik@gazeta.pl[1]; a.gil@ajd.czest.pl[2] ; nowackau@interia.pl[3]

Abstract: The following article presents the concept of uses virtual measurement tools as a modern means of teaching in the engineers' education, in the context of effectiveness of developing expected engineering skills. A short evolution of the measurement system leading towards visual programming is also presented. The article describes construction of virtual devices in LabVIEW, methodology of designing measurement systems in this program, as well as features of virtual instruments and their possible uses in laboratory measurements.

Keywords: virtual laboratory, visual programming, engineering education.

Introduction

Teaching technical subjects is specific - students should be given a chance to understand the subject, i.e. they should be taught to how use their knowledge in practice, therefore measurement laboratories are very important in engineering education.

Classic measurement laboratories, equipped with gauges (transducers) properly selected to the measured quantity (manometer, dynamometer, etc.), measuring instruments (oscilloscopes, voltmeters, frequency meters), as well as signal processing systems are increasingly becoming replaced by virtual laboratories.

"Virtual (Latin: *virtualis* – possible) means possible to come into existence or theoretically possible. It is something that does not exist in reality, but it fulfils its role by means or methods other than the traditional ones. Nowadays, the concept of being "virtual" is strictly related to computer technology, as: existing solely on a screen and in computer memory, but acting as in reality (Świsulski, 2005).

Virtual instruments allow for performance of actual measurements, however, with the use of other methods. In classic instruments, buttons and switches placed on the front panel are used. In a virtual device, the front panel with buttons, switches, etc. is simulated on the screen of a computer and operated by mice, keyboard or a touchscreen. (Jemielniak, 2012)

Modern metrology is explicitly based on methods which virtualise the act of measuring. Advantages of virtual measurement methods cause gradual replacement of classic methods. Virtual measurement instruments can be constructed at relatively low expenditures both for their use at test stations in laboratories educating engineers of various specializations and in scientific laboratories. As a result, scientists and engineers will be able to work on projects by conducting remote simulation of events, experimental data interpretations and in some cases, performance of actual experiments in a specially adopted instrumentation laboratory.

The aim of this article is to present the modern concept of uses virtual measurement tools in the engineers' education. The examples shown in the article allow to analyze and evaluate the usefulness of these tools.

Methodology

Computerization of the measurement process

Computerization of the measuring process consisted, at the beginning, in developing a control program from scratch with commands for communication between the computer and input/output devices. To this end, high-level programming languages were used (Basic, Pascal, C). This very laborious way allowed for receiving data from sensors, processing it and sending commands.

The next step was developing a standard which defined the methods for measurement devices programming. In the 1990s SCPI standard was established (Standard Commands for Programmable

Instruments). It was a set of instructions allowing for complete programming of measurement device regardless of its type.

The development of high level programming languages (Pacal or C) and the graphical operating system Windows at the beginning of 1990s set conditions for development of integrated measurement environments. Different companies (National Instruments, Hewlett-Packard, etc.) began to build programming packages, facilitating design of measurement systems of high measuring capacities and easy operation. Textual and graphical environments (e.g. LabWindows/ CVI) and those typically graphical (LabVIEW) were created. The former are a combination of graphic design with code writing, whereas the latter do not involve program code writing at all. An increasing number of technical universities has been actively engaging in this type of teaching. We can find several examples of the use of virtual laboratories in literature (Sudół, 2009; Urbański, 2010; Biernat, Urbański, 2012).

Purely graphical environments proved to be better received, whereas classic programming languages have been gradually landing on the scrap heap. Currently, a significant majority of measurement systems programming environments are graphical environments (Winiecki, Nowak, Stanik, 2001). They do not require the user to know any programming languages.

Programming in LabVIEW

The most popular among a few visual programming environments is LabVIEW (Laboratory Virtual Instruments Engineering Workbench) developed by National Instruments. It is the best developed and user-friendly environment with the largest libraries. (National Instruments, 2012)

The adopted terminology, graphical symbols and block method of presenting how the program functions in LabVIEW do not differ from the notation commonly used by engineers, while the "pictorial" character of the environment helps to graphically present data (Tłaczała, 2002).

Programming in LabVIEW involves the following steps:

- a control panel of the virtual device is designed in a panel window,
- a graphical program is created in a diagram window,
- an icon (plug) is designed to represent the program, enabling its attachment to other programs.

The panel constitutes an interactive interface with the user (Fig. 1). It simulates the front panel of the physical measurement device (Świsulski, 2005). Controls for entering data into the program (switches, knobs, displays) are placed on the panel along with Indicators used for exporting data from the program to the user (displays, charts, etc.). The panel elements are operated by a mouse, a keyboard or a touchscreen.

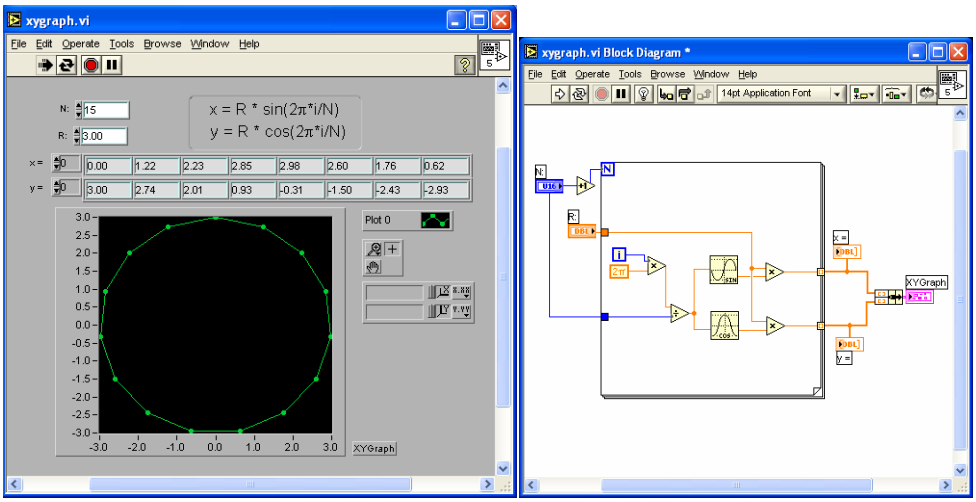

Figure 2. The front panel and the diagram in the LabVIEW environment. Program example (virtual instrument), which draws a polygon (Targowski, Sylwestrzak, Bajraszewski, 2012).

The diagram in figure 1 contains a source program of the application in a graphic language. The diagram shows icons which represent particular elements of the program (function blocks) and connections among elements indicating the data flow direction. Programs in LabVIEW are controlled by the data flow, which means that successive node of the block diagram starts to work when it has all of the required data on all of their inputs and finishes its work by generating data to all of its outputs. There are no limitations as far as sequential programming is concerned, function blocks may have many independent data inputs, many operations can be performed simultaneously.

Methodology of measurement system design

Methodology of designing a measurement system in LabVIEW can be divided into a few stages (Winiecki, Nowak, Stanik, 2001).

1. Introduction to a design:
   a) defining functions and parameters of the device being designed,
   b) selection of the right equipment (sensors, data acquisition devices, etc.),
   c) analysis of an equipment controller or designing a new controller.
2. Creating a graphical panel of the measurement device:
   a) placing elements which are necessary for functioning of the device inside the panel editor window, i.e.: knobs, slides, displays, etc.,
   b) proper distribution and configuration of panel elements.
3. Creation of a block diagram (graphic source code):
   a) distribution of the existing icons (they are automatically moved onto the diagram from the graphical panel which is being created - the icons are equivalents of panel elements),
   b) inserting additional elements and procedures (if necessary):
      – loops e.g.: while, case,
      – interface functions,
      – controllers of physical devices,
      – sequence structures, etc.
   c) combining elements so that the device functions properly.
4. Start-up and program testing.

Virtual measurement devices

Most of the traditional measurement devices can nowadays be replaced by a new system composed of a sensor, a signal preparation module and a computer equipped with a data acquisition card (DAQ) and suitable software (DasyLab, LabWindows/CVI, LabVIEW and other). This type of software helps to build virtual measurement and control devices. They use the vast and still increasing capacities of computers in calculations and presentations of results. Thus, in a virtual device a signal is provided not to the input of e.g. an oscilloscope, but DAQ cards of the computer in which the program performing the tasks of an oscilloscope, i.e. a “virtual oscilloscope” is. If we need a different device, e.g. a voltmeter, we shut down the oscilloscope program and start a programme which performs the tasks of a voltmeter or another device. If we do not have the program which would serve as the virtual device we need, we can write the program ourselves by using, e.g. the LabVIEW programming environment.

One of the most crucial advantages of a virtual device is its functional flexibility, i.e. one equipment block (a computer + a DAQ card) enables creation of a wide collection of various virtual devices performing different functions (Winiecki, Nowak, Stanik, 2001). This fact, as well as reducing the amount of equipment to the minimum have substantially decreased the cost of the device. What is more, the time spent on developing a new device and its modification in programming environments, such as LabVIEW, is very short.

A virtual device does not impose such limitations as it is in the case of a physical device, which provides only one displayer or scale on which all of the information is presented. Each measurement parameter can have a separate indicator in a virtual device. Therefore, operation of a virtual device is

intuitive and easy. It also ensures a lot freedom in shaping the front panel. You can freely distribute indicators, select their shape, style and size. The shape and size of displayers, colours of particular elements and background, as well as types of chart lines, etc. can be selected. Moreover, options for presentation of results are wider. These are no longer limited to single numbers, but instead include also tables, charts, matrices, graphs. Processing of results is not limited either, as it is in the case of a classic device. The software enables to extend processing and results analysis range.

Results and discussion

Virtual RLC series circuit

The below figures present a panel and a diagram (Fig. 2) of a virtual device, which simulates the functioning of a simple, perfect RLC series circuit (National Instruments, 2012). An experimenter may set the supply voltage frequency values, as well as resistance, inductance and capacitance. The device calculates current in the circuit and the values of current on particular elements of the circuit.

The front panel of the device contains the following parts:

- four pointers (Current, U_R Voltage, U_C Voltage, U_L Voltage) with additional numeric indicators;
- three numeric controls: R, L, C for setting the values of circuit elements;
- two numeric controls in the form of knobs (Voltage, Current) for setting the parameters of the supply voltage;
- a Stop button (end of simulation) used to stop the work of the program;
- graphical elements.

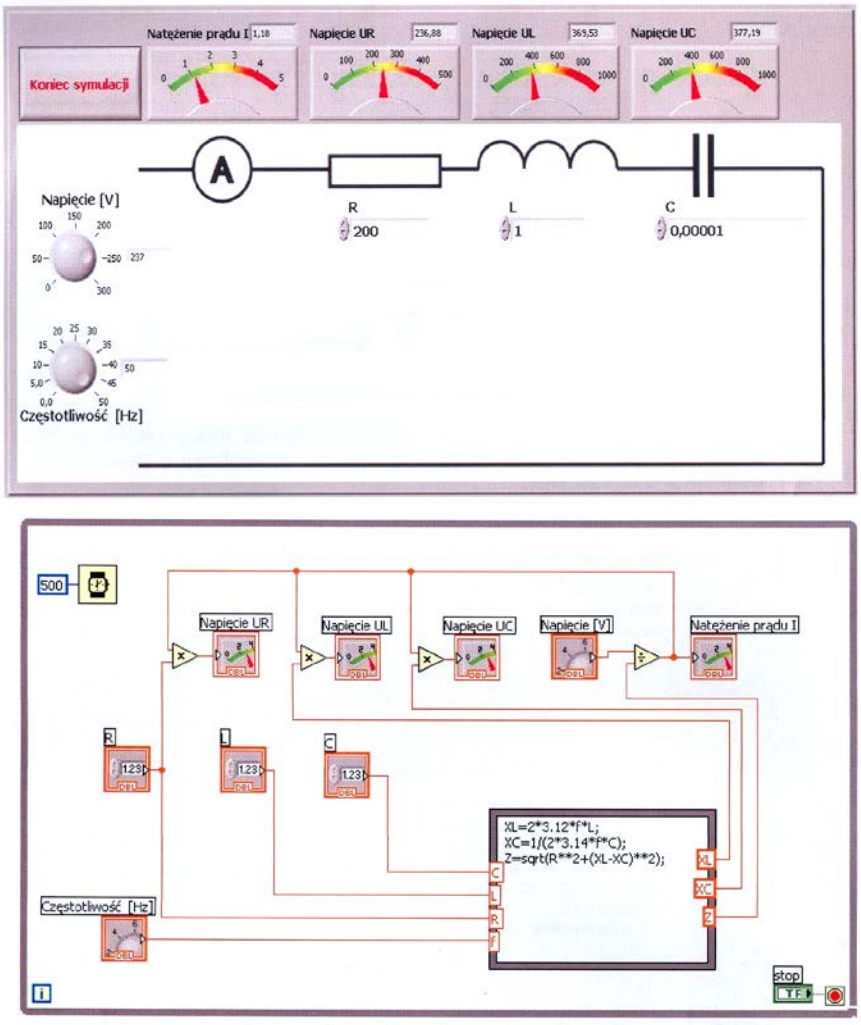

Figure 3. The front panel and the diagram of the virtual device.

Values set by the experimenter:

U - supply voltage in V (volt);
F - supply voltage frequency in Hz (hertz);
R – resistance of the resistor in the circuit Ω (ohm);
L – inductance of the coil in H (henry);
C – capacitance of the capacitor in F (farad).

Intermediate values and values to search for:

X_L – inductive reactance measured in Ω (ohm), $X_L = 2\pi fL$;
X_C – capacitive reactance measured in Ω (ohm), $X_C = 1/(2\pi fC)$;
Z – circuit impedance measured in Ω (ohm), $Z = R^2 + (X_L - X_C)^2$;
I - current in the circuit measured in A (ampere), $I = U/Z$;
U_R – voltage across the resistor R measured in V (volt), $U_R = I \cdot R$;
U_L – voltage across the inductance L in V (volt), $U_L = I \cdot X_L$;
U_C – voltage across the capacitor C in V (volt), $U_C = I \cdot X_C$.

The device diagram (Fig. 2) presents all of the elements placed on the panel. These include voltage, frequency, resistance, inductance and capacitance controls, V voltage and I current indicators, as well as U_R, U_L, U_C voltages. There is also a stop switch. Calculation of the required parameters X_L, X_C, Z is performed in the Formula Node structure. In order to ensure continuous operation of the program, a loop While has been introduced. A Wait module prevents monopolization of computer resources by a continuous loop.

ALPHA controller application

Another example of using visual programming is ALPHA controller from Mitsubishi (Fig. 3). It is a microprocessor device which can control various devices. It has 6 relay inputs (which assume the signal values of 0 and 1) and 4 outputs, therefore it can operate 6 input signals and simultaneously control four objects.



Figure 3. ALPHA logical controller, model AL-10MR-D (Automatyka Napędy..., 2012).

The controller is programmed with the use of function blocks, i.e. blocks which fulfil different functions. Apart from 6 logic functions, the controller also has 16 blocks fulfilling other functions, which are useful in controlling. Programming of the controller is performed on a computer connected to the controller by an adequate conduit.

The controller is programmed by a graphic program, AL.-PCS/WIN-E, working in the Windows environment. The platform used for controller programming is the so-called Function Block Diagram Base (FDB) (Fig. 4). It is a field on a computer screen with squares along the left edge which correspond to controller inputs and squares along the right edge which are outputs. To the left from the FDB field, there are icons of input devices (IN), output devices (OUT), logical function (LOG), other functions (FUN), as well as an I/O cable icon. When programming a controller we place particular elements (lamps, switches, heaters, diodes, pistons, etc.) at the inputs and outputs of FDB fields. We distribute the function blocks inside the FDB field. We connect the inputs with the function blocks and the blocks with the outputs. In order to do that, we use the "drag and drop" method - by clicking on a given icon and releasing the mouse button in the place where we want to place the icon. Designing a control system comes down to arranging "logical blocks" and interconnecting them. After designing the system, we test it by selecting the simulation mode. By clicking on the inputs we change the signal value from 0 to 1 and the other way round. Colours change on the screen. In places where the value is 1, the red colour appears. Inputs, outputs and cables with no signal remain blue. We may track the course of a signal through the entire control system - from the system input to the output.

If the control system we designed does not work as it should in the simulation mode, we return to the design mode. We make changes and test the system again in the simulation mode. If it works properly, we copy the control program from the computer to the controller and we test how the controller controls a physical object connected to it. Figure 4 presents the examples of control programs.

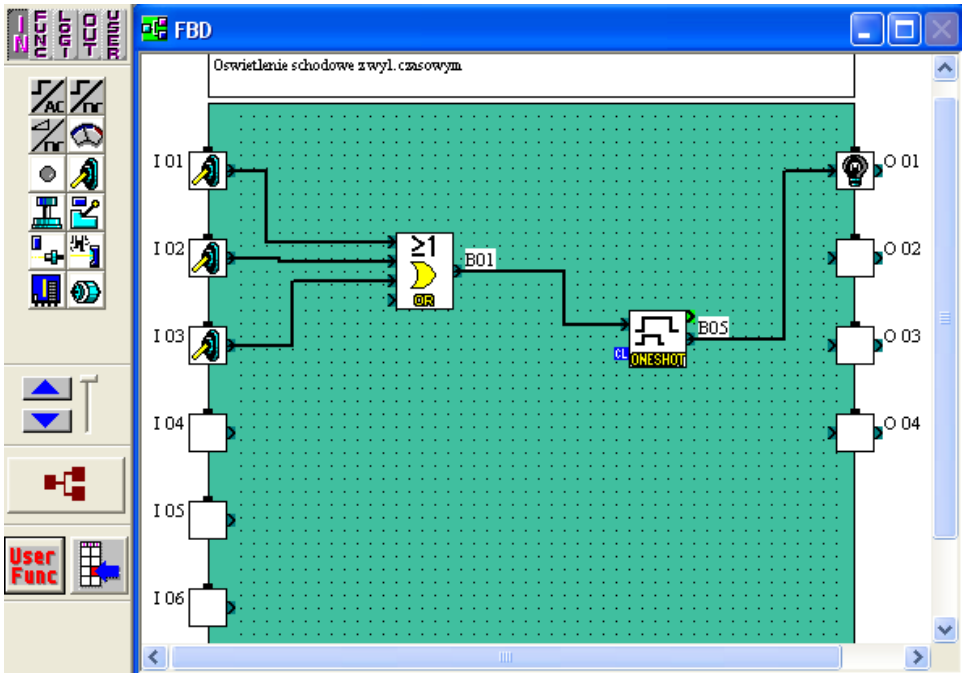

Figure 4. Function block diagram base.

Below we present a car park barrier control program (Fig. 5a, 5b). The barrier is activated and stopped by a signal from a remote control (input I01). The output signal which sets the barrier in motion is provided to the output O01. The same signal is simultaneously send through a delay unit and a pulse generator to output O02 where a warning lamp is placed. The delay unit and pulse generator trigger the blinking of the warning lamp during the barrier movement with the lamp reaction being delayed in relation to that movement. Symbolic photoelectric cells are marked on I03 and I04 inputs, which should be placed in front of and behind the barrier.

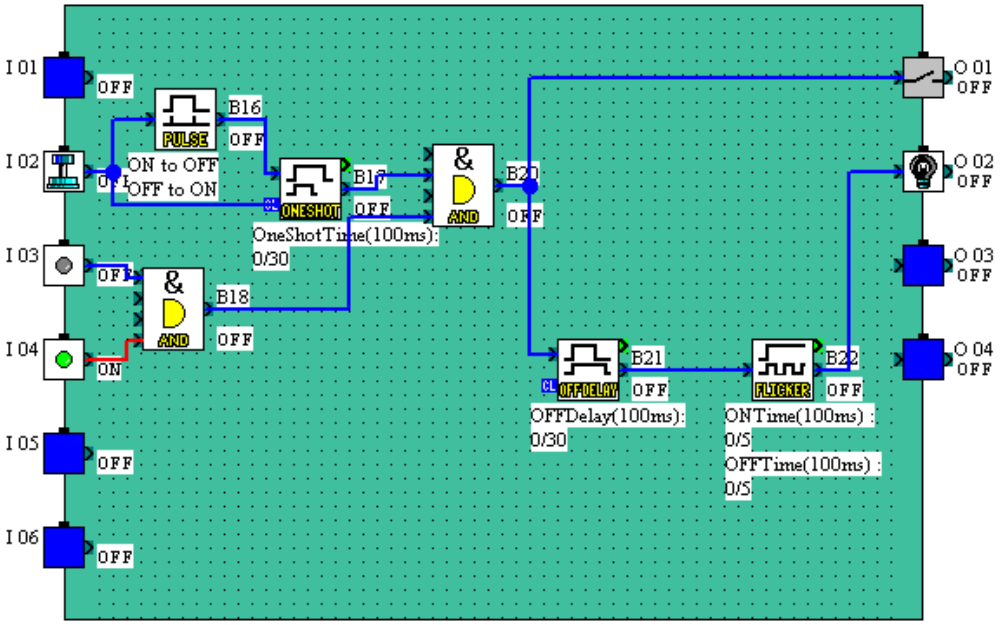

Figure 5. a) The car park barrier control program;

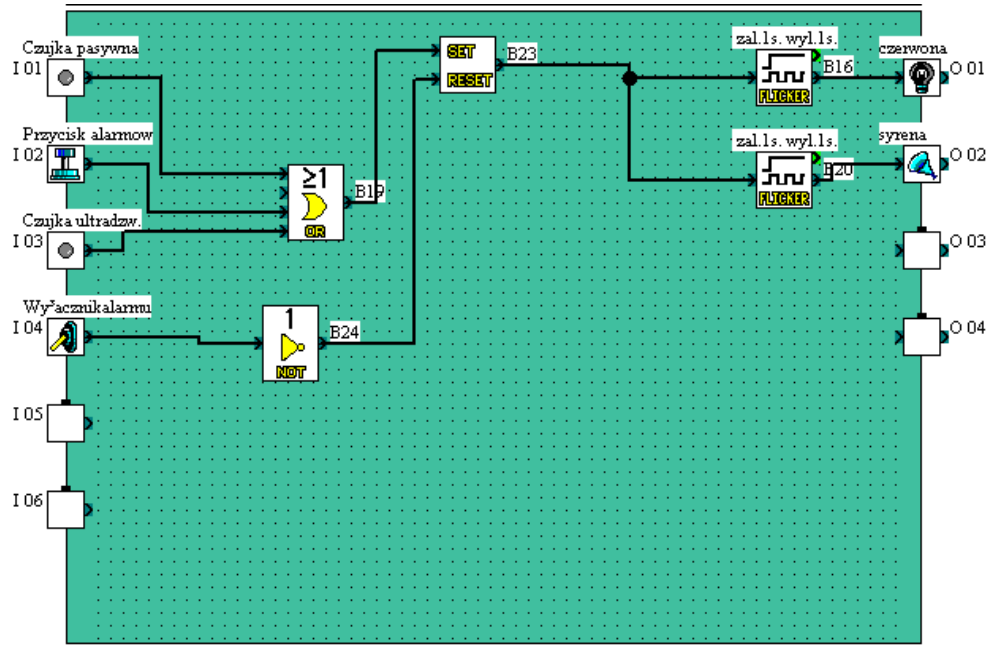

Figure 5. b) The alarm siren and lights control program.

They hold back the barrier when a car is within their reach. This prevents the barrier from damaging the car. Signals from those photoelectric cells reach the AND unit so that a signal from at least one of them could block the movement of the barrier. A result signal from photoelectric cells reaches the AND unit together with the signal from the remote control, since only a simultaneous signal from the remote control and both photoelectric cells may activate the barrier. A signal from the remote control reaches the PULSE unit earlier, which generates a unit pulse with the change of the remote control mode (from ON to OFF and vice versa) and to the ONE SHOT unit, generating a pulse with a programmed time, which enables closing and opening of the gate.

Conclusions

A modern didactic laboratory is a virtual laboratory, which replaces many (often very expensive) measurement devices. It is composed of a computer with adequate software, data acquisition card and appropriate sensors. It allows students to design and construct the required measurement devices on

their own. It combines educational, measuring and simulating aspects. Thanks to them, students become familiar with using modern research techniques and operation of integrated measuring systems in everyday work. An increasing number of technical universities have been actively engaging in this type of teaching.

An important obstacle which stands in the way of introducing the proposed methodology of technical classes is, unfortunately, a considerable amount of work, which is needed to design, make and test every laboratory station. Such effort can, however, be a part of studies curriculum. Performing the above-mentioned tasks may constitute the core subjects of Engineer's theses and - in the case of more advanced measuring systems - Master's theses. Such a solution regarding diploma papers bring many positive educational effects - it teaches students to be independent and to design with the use of modern IT tools.

Modification of measurement procedure by changing software installed on a computer, without a change of equipment components makes research and experiments more flexible, modern and easier for students of technical universities. Computer animation of a given phenomenon (technical process) combined with a simulation model allows students to conduct, with the use of computers, their own experiments and research, reflected in actual phenomena and processes, which is enabled by a well-equipped didactic laboratory. It should, however, be noted that computer simulation cannot and should not - despite its advantages mentioned in the article - entirely replace a physical laboratory or industrial practice by its virtual counterpart, but it may constitute complementary form of engineering education. Fascination with the computer world may not veil the fact that virtual laboratories are not sufficient in teaching all of the expected engineering skills, since they lower the feeling of hazard, introduce so many boundary conditions that the object of simulation does not often resemble its physical equivalent and a set of complex phenomena is thus treated cursorily.

Bibliography

1. Automatyka Napędy Komunikacja (Drives of Automation Communication), [online] [06.12.2012]. Available at http://www.plcs.net.pl/index.php/home.html (In Polish)
2. Biernat A., Urbański W. (2012). *Wirtualna technika pomiarowa w laboratoriach maszyn elektrycznych*, Napędy i Sterowanie - miesięcznik naukowo-techniczny, Wydawnictwo Druk-Art. SC, Nr 5, p. 80-87, Racibóż, Poland. (In Polish)
3. Jemielniak K., *Przyrządy wirtualne* (Virtual instruments), [online] [06.12.2012]. Available at http://www.cim.pw.edu.pl/labview/ (In Polish)
4. National Instruments, [online] [06.12.2012]. Available at http://www.ni.com
5. Sudół W. (2009). *Nowoczesna metrologia wirtualna, budowa stanowiska, do badań trójfazowej prądnicy synchronicznej* (Modern virtual metrology, construction position, to R-phase synchronous generator). Wyższa Szkoła Ekologii i Zarządzania (School of Ecology and Management), Warsaw, Poland. (In Polish)
6. Świsulski D. (2005). *Komputerowa technika pomiarowa* (Computer measurement technique). Agenda Wydawnicza "Pomiary, Automatyka, Kontrola", Warsaw, Poland. (In Polish)
7. Targowski P., Sylwestrzak M., Bajraszewski T. Środowisko LabView- własności i przykłady zastosowań (LabView environment - properties and application examples). Instytut Fizyki UMK, Optopol, [online] [06.12.2012]. Available at
8. http://dydaktyka.fizyka.umk.pl/komputery/pliki/Srodowisko_LabView.pdf
9. Tłaczała W. (2002). *Środowisko LabVIEW w eksperymencie wspomaganym komputerowo* (Środowisko computer-aided experiment with LabVIEW). Wydawnictwo Naukowo-Techniczne, Warsaw, Poland. (In Polish)
10. Urbański W. (2010). *Wirtualna technika pomiarowa w dydaktyce maszyn elektrycznych* (Virtual measuring technology in teaching electrical machines). Zeszyty Problemowe Maszyny Elektryczne (Electric Machines Exercise Problem), nr 87, p. 227-230, Katowice, Poland. (In Polish)
11. Winiecki W., Nowak J., Stanik S. (2001). *Graficzne zintegrowane środowiska pomiarowe* (Graphics Integrated in Test Environment). Wydawnictwo MIKOM, Warsaw, Poland. (In Polish)